Spatio-Temporal Consistent Mapping of Growing Plants for Agricultural Robots in the Wild

Luca Lobefaro Meher V. R. Malladi Tiziano Guadagnino Cyrill Stachniss

***Abstract*— Tracking changes in growing plants is important for automating phenotyping and robots managing crops. In this paper, we propose a system that uses a 3D model of plants along crop rows to enable a robotic platform to localize itself even in the presence of heavy changes and deforming the model to adapt the scene description to the new measurements. In particular, we focus on consumer RGB-D cameras due to their cost-effectiveness and ease of deployment on real platforms. Our approach exploits modern deep-learning-based feature descriptors and geometric information to obtain matches between 3D points corresponding to temporally distant sessions. We then use the associations in a non-rigid registration pipeline to obtain the final result, an updated representation of the 3D model that reflects plant changes. Using a standard RGB-D sensor, we validate our approach on a real-world dataset recorded in a glasshouse. We obtain accurate 4D models of the plants and track the plant traits' evolution over time. We show, through experiments, that our method is applicable to interpolate plant organs' evolution, a helpful result for phenotypic trait measurement. We see our approach as a relevant step toward 4D reconstruction for robotic agriculture in the wild.**

I. Introduction

Crop field management requires the analysis of phenotypic traits, such as leaf shape and area, and involves their measurement over time. Being able to automate this process is of fundamental importance in speeding up production times and in achieving a more sustainable crop production.

Several methods for automatic phenotypic traits estimation have recently been proposed [5], [14], [19], [40]. However, most of these methods assume to have an accurate 3D representation of the crop field, which is typically unavailable. Moreover, some of these approaches require invasive intervention to isolate plant parts. Such intervention is not always possible because, in most cases, plants cannot be removed from their place to be measured.

When operating in agricultural environments, robots typically build a new map every time they perform an inspection. This is necessary because plants change shape over time. Building a new representation implies no relationship with the previous one, preventing the possibility of tracking the evolution of plants' organs. A possible solution is to re-use the previous map, deforming it according to new measurements through non-rigid registration.

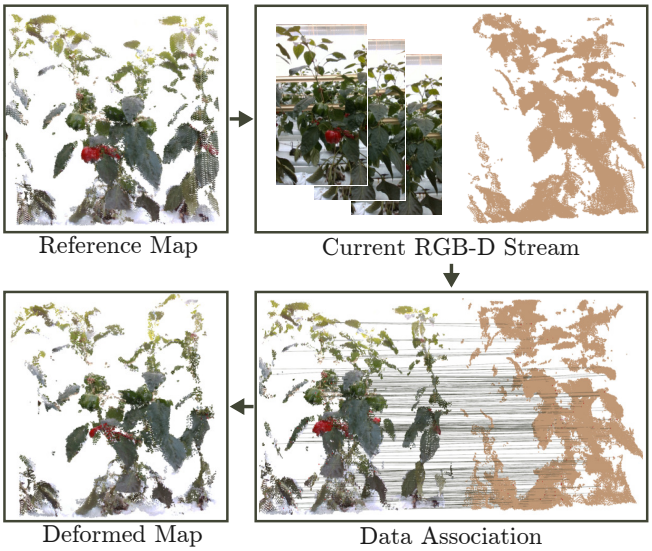

Fig. 1: We start with a reference 3D point cloud (top left). Given the RGB-D stream (top right) from a session recorded at a later time, we extract the depth cloud (in brown) and we compute associations between it and the reference point cloud (coloured). We use the matches (bottom right) to perform the non-rigid deformation of the reference map. The final result (bottom left) is a deformed point cloud which reflects the current observations.

Non-rigid registration has been discussed in the literature [6]. Despite that, most existing methods assume complete and well-defined shapes, usually coming from computer graphics setups. Working with robotics sensors means working with noise and incomplete shapes, introducing another level of complexity. Some studies have tried to address these issues [27], but none is specific for agriculture settings, where we have new challenges like dealing with highly repetitive scenes.

This paper proposes a mapping system that can reuse previously built 3D point clouds of agricultural environments and continue working with it through non-rigid registration. For this reason, our pipeline allows farmers to keep track of the evolution of plants' organs. To enable this, we also need to perform localization with respect to the previous model, even in the presence of plant growth and deformations. Our approach exploits visually stable features of plants to be robust to drastic environmental changes. Fig. 1 shows an example of the result achieved with our method.

The main contribution of this paper is a novel pipeline for spatio-temporal mapping of plants in the wild, relying only on an RGB-D camera stream. Our approach can (i)

All authors are with the Center for Robotics, University of Bonn, Germany. Cyrill Stachniss is additionally with the Lamarr Institute for Machine Learning and Artificial Intelligence, Germany.

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).

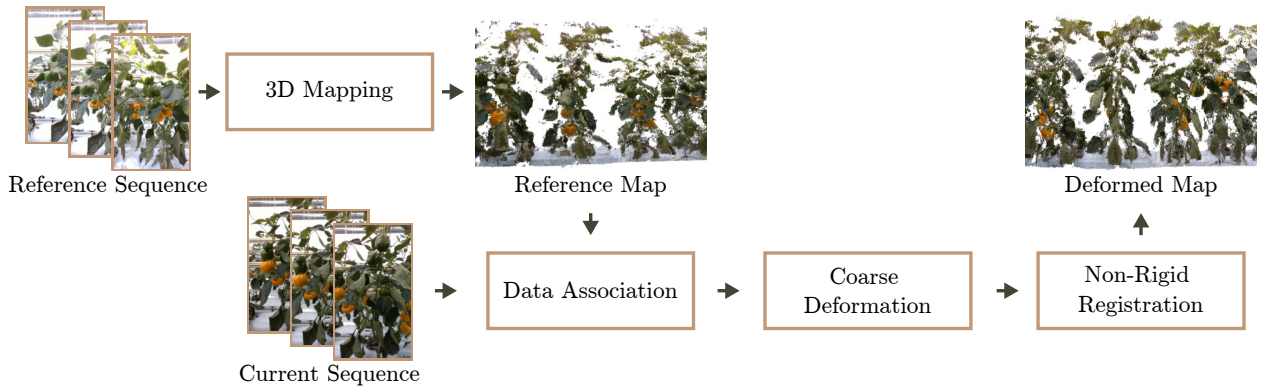

Fig. 2: Our pipeline for 4D reconstruction. First, we produce a 3D point cloud from the reference sequence. Then, we associate points between the reference map and the cloud of the current depth stream. We finally use these associations to deform the map, accounting for large deformations. A non-rigid ICP algorithm is applied to fine-tune the result, accounting for small deformations. The output is the reference map deformed to reflect the changes.

reuse and update previously built maps, (ii) enable to track the evolution of plants' organs over time, and (iii) works while being robust to sensor noise that characterizes consumer cameras. These claims are backed up by the paper and our experimental evaluation. Our approach is as an essential step towards 4D reconstruction and shape prediction for robotic agriculture. The open-source implementation of our approach is available at: `https://github.com/PRBonn/spatio-temporal-mapping`.

II. Related Work

Robots in agriculture is an active research field nowadays [21], [39]. Challenges imposed by population growth and recent events, like the COVID-19 pandemic in the early 2020s, caused issues and affected the world food production. Advanced robotics solutions have the potential to increase production and ensure an agricultural economy robust to pandemic emergencies [9], sustainably in the long run [1].

Obtaining 3D representations of plants is central in the context of robotics for agriculture. Nevertheless, it is challenging due to the complex structure and the dynamic nature of the environment. Furthermore, measurements are performed mainly outdoors, where wind and other factors can push the current state-of-the-art systems for mapping and SLAM to their limits [31]. Recently, Islam et al. [16] proposed a system for stereo visual SLAM that outperforms state-of-the-art methods on the agriculture ROSARIO dataset [24]. They show how other visual SLAM methods fail in agricultural environments. Then, they propose an image enhancement technique for visual features recovering that makes their system robust to low-light and hazy scenarios. Ding et al. [8] offer a survey on recent developments in localization and mapping in agriculture.

The 3D models of plants are only one part of the story as methods for precision agriculture also require temporal information [23]. These are needed to keep track of the evolution of plants' organs and for fruit growth analysis [22]. Many methods are analyzed both in the computer vision and the robotics communities. Chebrolu et al. [3] address the problem of spatio-temporal plant traits tracking. They exploit the skeletal structure of plants to perform 3D point registration. Demby's et al. [5] propose a system for bundle registration of 3D models of plants. They account for deformations caused by growing or motion resulting from outdoor forces. In particular, they propose an objective function to optimize and align point clouds at different growing stages. Heiwolt et al. [14] set out an approach for identifying and tracking the individual plant components over time. A leaf-shape compression pipeline allows encoding this information and quickly comparing plant organs to recognize the same one at different growing stages. Xiang et al. [40] suggest a two-step approach for plant growth tracking. First, they perform a spatio-temporal registration of plant point clods, and then, they compute a cost correlation matrix for single organ association.

All these approaches work under two constraints. First, a precise, high-resolution point cloud captured with LiDAR is available and second, a single plant instance is observed. The Pheno4D dataset proposed by Schunck et al. [29] is a prominent example for that. Unfortunately, in most real settings, obtaining such measurements is challenging because isolating single plants is only possible by changing the environmental structure and submillimeter precise poses and measurements are rarely available.

Non-rigid registration is a known problem in computer graphics, computer vision, and robotics [6]. The goal is to compute an alignment between a given object, represented as a mesh or point cloud, and a target object whose difference is in a non-rigid deformation. Most methods exploit the same idea of rigid ICP and solve the problem with iterative matching. An example is given by Amberg et al. [2]. Glira et al. [11] propose the usage of piece-wise tricubic polynomials to move a step further. This reflects in more flexibility and higher efficiency. Recently, deep learning has also been exploited for this purpose; examples are FlowNet3D [17] and HPLFlowNet [12]. In [18], we have tackled the task of estimating data associations across mapping sessions. Registration, however, was not part of the work.

With larger deformations, iterative methods with nearest neighbours search tends to fail. Helpful are pre-computed point-to-point associations exploiting geometric or visual features. They allow for handling larger deformations. In the literature, this problem is known as "coarse registration". An example is the classic work by Sommer et al. [34]. They propose to approach the deformation as an optimization problem. In particular, first, they compute a deformation graph that embeds the underlying object. Then, they launch an optimization to find the optimal affine transformations attached to each node of the graph, trying to preserve local shapes. In this specific work, Sommer et al. [34] perform 3D point matching by hand. In other works, the matching is obtained using orientation-invariant point descriptors. The most relevant works on 3D feature descriptors for automatic point matching are FPFH [26] and SHOT [28]. Examples of 3D keypoint detection are NARF [33] and 3D-SIFT. The latter is an adaptation of the SIFT algorithm for 2D keypoint detection by the community of the Point Cloud Library [25]. Matching 3D descriptors with classic methods introduces errors. For this reason, methods for matching pruning have been explored by Tam et al. [35].

Although several non-rigid registration methods have been proposed to work with noisy RGB-D sensors [15], [20], [42], none explore the same setting in agriculture. Furthermore, their approaches aim to produce a dynamic representation of moving scenes, a goal that differs from the one proposed in this paper. We assume each scene to be static in each session, with deformations happening only between different recordings. This is a realistic scenario when dealing with plants that are recorded by a robotic platform.

We propose a pipeline for 4D mapping in agriculture settings under growth and deformations of the crop map. We only rely on consumer RGB-D data sensors as input, allowing our system to be easily adopted on real robotic platforms. We exploit previous knowledge of plant structures to localize the current RGB-D stream on a previously built map. Then, we compute point-to-point associations using neural visual features. Finally, we start a coarse non-rigid registration followed by a final refinement through non-rigid ICP. Our experiments on a real robotic platform [32] show that this approach can produce robust 4D reconstruction even with incomplete and noisy shapes.

III. Our Approach

We propose a pipeline that generates a spatio-temporal consistent model of plants. In particular, our proposed system, starting from a 3D point cloud of plants, can deform the 3D scene according to new measurements performed sometime later. This is possible by 3D point matching between the cloud and the current RGB-D stream, using a method to find data association across time similar to our previous work [18]. For point matching to be possible, we need to solve the visual odometry of the new frames on the previous representation. Once we obtain the matches, we can deform the 3D clouds. The result is a representation that reflects the current state of the plants.

Fig. 3: Left: example of camera point of view of the dataset used. Right: point cloud generated. In the square: the part of the plant used as stable feature.

To develop the pipeline, we use data recorded with the robotic platform introduced by Smitt et al. [32], in which vertically mounted RGB-D cameras can capture side-views of sweet peppers rows in a glasshouse.

A. 3D Mapping of Plant Rows

The first step of our pipeline is to produce a 3D point clouds $\mathcal{M}_r$ representing rows of sweet peppers in the glasshouse from a sequence of RGB-D images. This is useful in the first session, when the goal is to produce the reference map that is then updated in the next sessions. Let us define the sequence of images as $\mathcal{S}_t = \{\mathcal{I}_t^1, \mathcal{I}_t^2, \ldots, \mathcal{I}_t^n\}$, where n is the number of images recorded in the session at time t. For each image, we first generate a coloured 3D point cloud using the classic pinhole camera model [13], filtering out all those points that are outside the relevant depth range as suggested by Smitt et al. [32].

We assume that the first pose of the robot is at the origin of our reference system. After that, for each incoming frame, we use the constant velocity model [36] as an initial guess for the next pose. It is a reasonable assumption in our setting because the robot moves in the glasshouse roughly at constant velocity with a relatively slow speed along the rows. Then, we refine the pose using the plane-to-plane ICP approach proposed by Segal et al. [30], which can compute a precise pose even with relatively small plant motion, such as leaf vibration due to wind. Following the insights of Vizzo et al. [37], we use two downsampled versions of the point cloud extracted from the RGB-D images: the first one, voxelized with a voxel size of $0.01\,\text{m}$ is used for local registration, and the second one, voxelized with a voxel size of $0.001\,\text{m}$ is denser, and we use it to produce the final point cloud. Along the same line as Vizzo et al. [37], we do not use the centre of each voxel as a representative for it, as in most approaches in the literature; instead, we select one point for each voxel avoiding the introduction of discretization noise.

B. Visual Odometry

At this point, we have a 3D point cloud representing the plants as they were in the last mapping session. The

goal is to update it once we visit the same place sometime later in a subsequent session, for example, a week later. Between the two sessions, plants grew and changed shape. The goal is to deform the first representation to match these changes and update it with the new data. To do that, we need, for each frame in the new sequence, to understand which part of the field we are observing. Then, we can use this information to keep track of changes. We approach this problem with frame-to-frame visual odometry. This allows us to compute each frame's pose according to the reference map. In particular, let us define the new sequence of images as $\mathcal{S}_{t+1} = \{\mathcal{I}^1_{t+1}, \mathcal{I}^2_{t+1}, \dots, \mathcal{I}^n_{t+1}\}$. This sequence measures the same row in the glasshouse recorded in $\mathcal{S}_t$. To avoid dependencies from a global localization module, we do a valid assumption: the frame $\mathcal{I}^1_{t+1}$ has the same pose as the frame $\mathcal{I}^1_t$, this because the robot starts the process of each row from the same position, corresponding to the beginning of the glasshouse's row. Then, we will use the pose associated with $\mathcal{I}^1_t$ as an initial guess for the first frame in our localization pipeline.

Performing visual odometry with new images on an outdated 3D representation, such as one recorded a week ago, is problematic. The plants, in the time between the two sessions, have changed their appearance by growing or missing parts such as fallen leaves or harvested fruits. For this reason, it is not possible to use classical methods without introducing errors. The way we considered the best to approach the problem is to use the stable features of the environment as the only valid information for the odometry. In particular, we choose the plants' stems, as shown in Fig. 3. These, are extracted by applying a threshold on the z-axis on the reference map.

The operations carried out are as follows. First, we isolate the stable features. Then, we extract the 3D points for each incoming frame using the camera model and the depth information. We use the ICP algorithm to obtain the corresponding pose with respect to the map using the constant velocity model on the previous frame [36] as an initial guess. As proposed in KISS-ICP [37], we also utilize only a subset of the map for the registration to speed up the procedure and avoid memory issues. In this case, we also use two map versions with different voxel sizes, as explained in the previous section for mapping.

C. 3D Point Matching

Now that we have a pose associated with each frame $\mathcal{I}^i_{t+1}$ of the current sequence $\mathcal{S}_{t+1}$, we can extract helpful information to update our reference map. We use each frame's pose to extract the corresponding point cloud $\mathcal{P}^i_{t+1}$ employing the depth image. Then, we perform feature matching between the map $\mathcal{M}_r$ and $\mathcal{P}^i_{t+1}$ using visual and geometric information. In particular, we follow the idea described by Lobefaro et al. [18], which proved robust even with heavy geometric changes and plant deformations. The computed matches indicate how plants have evolved between the two sessions.

In order to use such an approach, we need an image of the first sequence $\mathcal{S}_t$ associated with the current frame $\mathcal{I}^i_{t+1}$. Unlike the original method, we eliminated the dependency on the visual place recognition module. This is possible thanks to the assumption made on the first pose (we always start from the same position every new session) and the sequential nature of our system. If starting from an arbitrary pose, an initial place recognition is needed [38] as used in out prior work [18] for initialization. To obtain the pairs of images, we take the pose of the current frame and search, among $\mathcal{S}_t$, for the one with the closest pose. Then, we exploit the coupled images $\mathcal{I}^i_{t+1}$ and $\mathcal{I}^j_t$ to compute keypoints associations between them. In particular, we use neural visual local features extracted with SuperPoint [7], and we apply a RANSAC schema [10] to filter outliers, using the information coming from the corresponding homography.

At this point, we filter the associations using geometric information. In particular, for each keypoint match $(\boldsymbol{p}_{r_i}, \boldsymbol{p}_{q_j})$, respectively in $\mathcal{M}_r$ and $\mathcal{P}^i_{t+1}$, we compute the corresponding 3D points using the pinhole camera model. Then, we search for the nearest neighbour in both point clouds. If we find a 3D point under a threshold in both maps, we have a point match in the cloud; otherwise, we discard this association. In this way, we filter out all those associations corresponding to part of the image not belonging to the plants.

Another difference from the original method is that we use an adaptive threshold to discard outlier matches. In particular, once we determine all the associations, we compute the threshold as:

$$\delta_{\text{matches}} = \mu + \sigma \,, \tag{1}$$

where μ is the average distance of the matched points (with the maps in a common reference frame), and σ is the corresponding standard deviation. In this way, we discard all the associations that are comparably far away from the average distance for the current frame. This allows us to ignore outliers from bad SuperPoint associations.

Finally, we have as output a set of matches between the reference map $\mathcal{M}_r$ and the cloud of the current frame $\mathcal{P}^i_{t+1}$. Each association carries the information about how the point should be located after the deformation that occured between the two sessions.

D. Non-Rigid Deformation

The computed matches between the reference and the new sequence $\mathcal{S}_{t+1}$ are representative of how the plants evolved between the sessions. The last step is to update the reference map to reflect the evolution. We address this problem by computing a deformation of the reference point cloud, guiding it through the associations. We call this step of our pipeline coarse deformation because its goal is to address larger deformation. Finally, a non-rigid ICP algorithm is applied to refine the result.

We follow the idea by Sumner et al. [34], who propose to warp the reference map using associations as constraints. This allows us to have a deformation that accounts even for large changes. Afterwards, a non-rigid ICP algorithm

can be applied to fine-tune the resulting deformation. In the following, we briefly explain how this method works and provide the parameters used.

The first step is to compute the deformation graph of the map [34]. Instead of computing the deformation on the whole cloud, we voxel downsample the original cloud with a voxel size of $0.025\,\mathrm{m}$ to reduce complexity. We define the deformation graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ as the set of nodes $\mathcal{N}$ and edges $\mathcal{E}$. We use the voxel center to define the nodes and connect each of them to their $n = 6$ nearest neighbours with an edge. After computing the deformation, we propagate the information to all the points $\boldsymbol{p}_i$ in the original point cloud. We do this by taking the nearest node $\boldsymbol{n}_k \in \mathcal{N}$ of each point $\boldsymbol{p}_i$, and compute the new position $\tilde{\boldsymbol{p}}_i$ as a weighted sum of the set of nodes $\mathcal{N}_k$ that shares an edge with $\boldsymbol{n}_k$ ($\boldsymbol{n}_k$ included):

$$\tilde{\boldsymbol{p}}_i = \sum_{\boldsymbol{n}_j \in \mathcal{N}_k} w(\boldsymbol{n}_j, \boldsymbol{p}_i)(\boldsymbol{R}_j(\boldsymbol{p}_i - \boldsymbol{n}_j) + \boldsymbol{n}_j + \boldsymbol{t}_j)\,, \tag{2}$$

where $\boldsymbol{R}_j$ and $\boldsymbol{t}_j$ are the rotation and translation computed on the node $\boldsymbol{n}_j$ after the deformation. We precompute the weights for each vertex according to:

$$w(\boldsymbol{n}_j, \boldsymbol{p}_i) = (1 - \|\boldsymbol{p}_i - \boldsymbol{n}_j\| / d_{\text{max}})^2\,, \tag{3}$$

where d_{max} is the distance to the $(n+1)^{\text{th}}$ nearest node of $\boldsymbol{p}_i$. The deformation is solved as an optimization problem with the following energy function:

$$E = w_{\text{rot}} E_{\text{rot}} + w_{\text{reg}} E_{\text{reg}} + w_{\text{con}} E_{\text{con}} \tag{4}$$

In particular, following the insights of Chen et al. [4], the term E_{rot} minimizes stretching by biasing the solution towards isometry:

$$E_{\text{rot}} = \sum_{\boldsymbol{n}_k \in \mathcal{N}} \left\| \boldsymbol{R}_k^\top \boldsymbol{R}_k - \boldsymbol{I} \right\|_F^2\,, \tag{5}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The regularization error E_{reg} forces the local shape of the underlying structure to be consistent. We achieve it by summing the squared distances between each node's transformation applied to its neighbours and the actual transformed neighbour position:

$$E_{\text{reg}} = \sum_{\boldsymbol{n}_k \in \mathcal{N}} \sum_{\boldsymbol{n}_j \in \mathcal{N}_k} \left\| \boldsymbol{R}_k(\boldsymbol{n}_j - \boldsymbol{n}_k) + \boldsymbol{n}_k + \boldsymbol{t}_k - (\boldsymbol{n}_j + \boldsymbol{t}_j) \right\|_2^2\,, \tag{6}$$

We use the last term E_{con} to force the deformation. For each match obtained with our pipeline, we set:

$$E_{\text{con}} = \sum_{(\boldsymbol{q}_r, \boldsymbol{q}_q) \in \mathcal{M}} \left\| \boldsymbol{q}_r - \boldsymbol{q}_q \right\|_2^2\,, \tag{7}$$

where $\boldsymbol{q}_r$ is the point of the match in the reference map, $\boldsymbol{q}_q$ is the corresponding match obtained from the new measurement and $\mathcal{M}$ is the set of matches. With these residuals, we can perform the deformation, preserving the local shape and obtaining consistent warping. We use the following values for the weights: $w_{\text{rot}} = 1.0$, $w_{\text{reg}} = 10.0$ and $w_{\text{con}} = 10.0$.

Parameter	**Value**
Mapping Voxel Size	1 mm
Registration Voxel Size	1 cm
Depth Min Threshold	40 cm
Depth Max Threshold	1 m
Stable Features Min Threshold	0.0 m
Stable Features Max Threshold	1.2 m
Deformation Graph Connectivity	6
Deformation Graph Resolution	25 cm

TABLE I: All parameters of our approach.

Once we perform the deformation using this method, we also apply a non-rigid ICP approach. This is useful to fine-tune the result. In particular, we use the warp field estimation module proposed by Zampogiannis et al. [42], using the implementation that they offer with their point cloud processing library [41]. The result is an updated version of the reference map that reflects the current state of the plants.

IV. Experimental Evaluation

The main focus of this work is a novel pipeline for spatio-temporal mapping of sweet pepper plants. Our work allows to reuse and update a previously built map to match new observations made some time later, relying only on an RGB-D camera stream. The goal is to account for non-rigid deformation that plants undergo over time. This allows to track the evolution of plants' organs between different recording sessions.

We present our experiments to show the capability of our system to perform spatio-temporal mapping of plants on a real agricultural glasshouse dataset, with highly repetitive scenes and non-rigid changes. In particular, we approach the evaluation as a non-rigid registration problem. We evaluate the result by determining the overlapping between the result of our system and a ground truth representation. We compare our system with the warp field estimation module proposed by Zampogiannis et al. [42] and publicly available in the point cloud processing library presented in [41]. In the following, we explain how we collected the data on which we performed the evaluation and extracted the ground truth.

A. Data Collection

To collect our data, we used the robotic platform described by Smitt et al. [32]. It operates in a glasshouse in Bonn, Germany, for growing sweet peppers. We used a Intel RealSense D435i RGB-D camera to capture sweet pepper plants in an intermediate growth stage in a span of one month. For our purpose, we utilized only the middle camera because it already allows us to see the whole plant at this growth stage, as shown in Fig. 3. In particular, we collect seven datasets from June 20th 2023 to July 14th 2023. We perform the recording for each session on three different rows of the glasshouse. The plants' shape has changed drastically between the sessions; fruits have been ripened and changed colour, and some have been harvested. These conditions make our dataset very challenging for 4D reconstruction,

		Query																	
			2.			3.			4.			5.			6.			7.	
Ref		*B*	*C*	*C+F*	*B*	*C*	*C+F*	*B*	*C*	*C+F*	*B*	*C*	*C+F*	*B*	*C*	*C+F*	*B*	*C*	*C+F*
	r3	45.3	36.2	**46.4**	–	–	–	34.3	23.6	**35.9**	34.0	6.6	**36.4**	31.6	6.1	**32.6**	**29.9**	5.1	28.5
1.	r4	54.4	45.0	**55.4**	46.3	31.6	**48.4**	46.7	35.2	**48.5**	40.8	23.9	**42.9**	**39.6**	20.6	33.8	–	–	–
	r5	39.5	30.2	**40.6**	–	–	–	37.2	26.3	**39.4**	**32.2**	21.2	31.2	**29.2**	8.8	28.2	**32.5**	16.8	32.0
	r3	–	–	–	–	–	–	39.2	27.0	**40.0**	37.0	20.9	**39.6**	**33.4**	13.7	32.8	**28.9**	3.3	27.4
2.	r4	–	–	–	50.3	32.3	**51.3**	48.8	36.0	**50.0**	43.9	24.2	**43.9**	42.0	21.6	**42.9**	–	–	–
	r5	–	–	–	–	–	–	38.7	29.4	**41.0**	34.5	25.4	**36.7**	31.6	20.0	**32.4**	34.8	22.0	**36.8**
	r3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
3.	r4	–	–	–	–	–	–	51.4	39.4	**53.0**	46.0	28.5	**47.5**	43.7	27.4	**45.8**	–	–	–
	r5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	r3	–	–	–	–	–	–	–	–	–	39.0	26.1	**40.4**	36.7	24.2	**37.7**	35.3	20.8	**37.8**
4.	r4	–	–	–	–	–	–	–	–	–	47.7	32.0	**48.8**	43.8	27.2	**46.2**	–	–	–
	r5	–	–	–	–	–	–	–	–	–	37.1	27.9	**39.1**	32.5	24.0	**35.4**	35.5	22.2	**37.8**
	r3	–	–	–	–	–	–	–	–	–	–	–	–	42.7	30.6	**44.4**	39.4	23.0	**41.5**
5.	r4	–	–	–	–	–	–	–	–	–	–	–	–	48.8	35.2	**50.9**	–	–	–
	r5	–	–	–	–	–	–	–	–	–	–	–	–	39.7	29.3	**40.8**	40.4	31.3	**42.7**
	r3	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	41.3	28.5	**43.7**
6.	r4	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
	r5	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	42.7	33.4	**43.7**

TABLE II: Evaluation results. The values represent the fitness (higher is better) beetween the reference map (left column) and the current one (first row) that we indicate as "query", computed as an average of 5 runs. All the results are expressed in % and the best results are outlined in bold. The numbers from 1 to 7 correspond to different sequences recorded on different dates. In particular: 1 is June 20th, 2 is June 22nd, 3 is June 27th, 4 is June 30th, 5 is July 07th, 6 is July 11th and 7 is July 14th. For each date different rows of the glasshouse have been recorded. These are indicated in the second column. In particular, r3 correspond to row 3, r4 to row 4 and r5 to row 5. The columns corresponding to B represent the results of the baseline, C of the coarse registration only, C+F of the coarse registration + non-rigid ICP (see Sec. IV-C). Our approach combining coarse deformation and fine-tuning works better in almost all cases, especially when the temporal distance between reference and query is around one week. With higher distances the baseline has better fitness values, because it relies only on nearest neighbors. This results in higher fitness but worst organs association (see Sec. IV-D).

allowing the possibility to test the proposed pipeline in a real-world setting.

B. Ground Truth Data Generation

We evaluate our system as a non-rigid registration problem. For this reason, we need a ground truth that represents the real map after the deformation. Furthermore, we want it to be aligned with the reference map so that each plant has the same global position in both representations. To generate such a ground truth, we operated as follows. First, we generate our reference map $\mathcal{M}_r$ from the RGB-D stream of the reference session. Then, we use the RGB-D stream of the new session to compute a map $\mathcal{M}_q$ that is aligned with the previous one. In particular, we perform frame-to-map odometry, obtaining a pose for each image in the new sequence that is aligned with the reference map. With these poses, we use the approach presented in Sec. III-A to generate the map $\mathcal{M}_q$. The result is aligned with the reference map thanks to the odometry and represents the environment as it is in the new session. In Fig. 4, we show an example of ground truth maps (in brown) together with reference maps (coloured). It is easy to see that each plant's base is aligned with the same instance in the reference map.

C. Quantitative Evaluation

To evaluate the quality of our system, we treat the problem as a classic non-rigid registration problem. Given the reference map $\mathcal{M}_r$ and the map obtained from the new session $\mathcal{M}_q$ (as explained in the previous section), our pipeline will deform $\mathcal{M}_r$ in such a way as to reflect the changes that occurred between the two sessions. The result will be a new map $\tilde{\mathcal{M}}_r$ obtained by deforming $\mathcal{M}_r$. The evaluation is carried out by computing the fitness between $\tilde{\mathcal{M}}_r$ and $\mathcal{M}_q$:

$$\text{fitness} = \frac{\#\text{ inlier correspondences}}{\#\text{ points in } \mathcal{M}_q} \cdot 100\,, \tag{8}$$

where the inlier correspondences are computed with the nearest neighbour search inside a sphere of $0.004\,\text{m}$ between points in $\tilde{\mathcal{M}}_r$ and $\mathcal{M}_q$.

In Tab. II, we show the values for the proposed metric. Each row of the table represents a reference map recorded in different sessions. Each column represents the sequence used to deform the reference, which we indicate here as "query." For example, the cell in the first row and column 2 indicates the fitness value obtained deforming the map recorded in June 20th using the RGB-D stream of June 22nd. The three different values indicate the result obtained with the baseline (*B*), the one obtained using only the coarse deformation (*C*), and lastly, using our complete approach (*C+F*). Our complete pipeline (columns *C+F* in the table) gives better results in almost all cases, especially with one week of difference between reference and query map. The baseline performs better with larger time distances. Applying the coarse deformation alone does not give good results. We will explain this behavior in the following paragraphs.

The baseline is a non-rigid ICP approach. It brings each point in the reference closer to the nearest one in the query, while maintaining local consistency. This results in higher

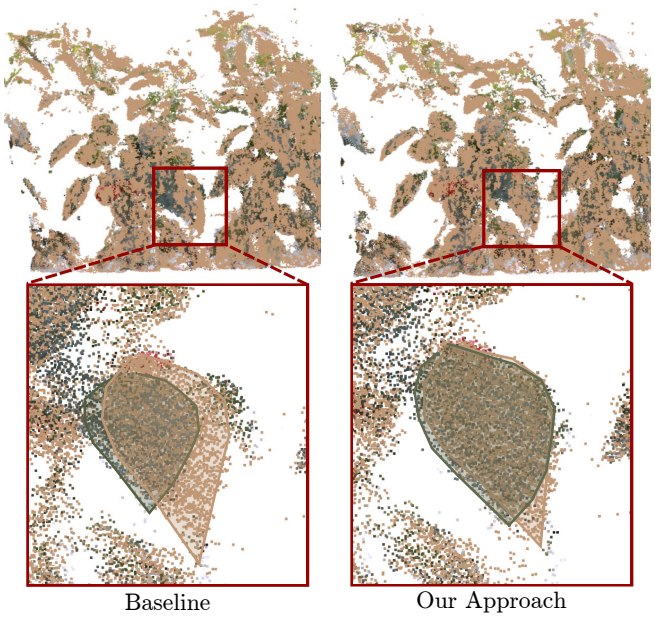

Fig. 4: The brown cloud represents the ground truth, the green cloud represents the deformed reference map. In the red squares we outline the visual differences between the results. We drew the contours of the leaves on the clouds to help visualization. In particular, with our approach, the leaves are in the correct position after the deformation, while the baseline deforms them to match as much as possible to the neighbours, without considering any other information.

fitness values even if the nearest points do not belong to the same plants' organs in both maps. With the coarse approach, we aim to bring each plant's organ nearest to its new position in the query map, for example, translating, rotating, and resizing a leaf in its new position. This is still not enough for a complete overlap between the two maps, as fitness values show. To obtain that we need to refine the result using non-rigid ICP. In this case, we can ensure that the nearest points computed with ICP belong to the same plants' organs in the two maps because now they are closer thanks to the previous coarse registration. We present a qualitative result in the next section to better show this. We also outline that the baseline requires a complete map of the current session to deform the reference, while the coarse approach requires only the RGB-D images. This can be exploited in a real-time pipeline for deformation.

D. Qualitative Evaluation

In the context of non-rigid registration, looking only at the numbers can give us only a partial judgment of the quality of the results. Furthermore, the metric used does not give us information about the validity of the final shapes. Since the ultimate goal of the work is to produce a representation consistent with what is observed, a qualitative assessment of the point clouds is also essential. For this reason, in Fig. 4, we show a visual representation of the results obtained, comparing them with the baseline output. We outline the leaves' contours to improve visualization. While the baseline tends to shrink the reference leaf (green in the picture) to the nearest points in the ground truth cloud (brown in the picture), our approach first moves the leaf to the correct position, stretching it to match the new dimensions. Then, a final refinement using nearest neighbors allows us to align the two clouds.

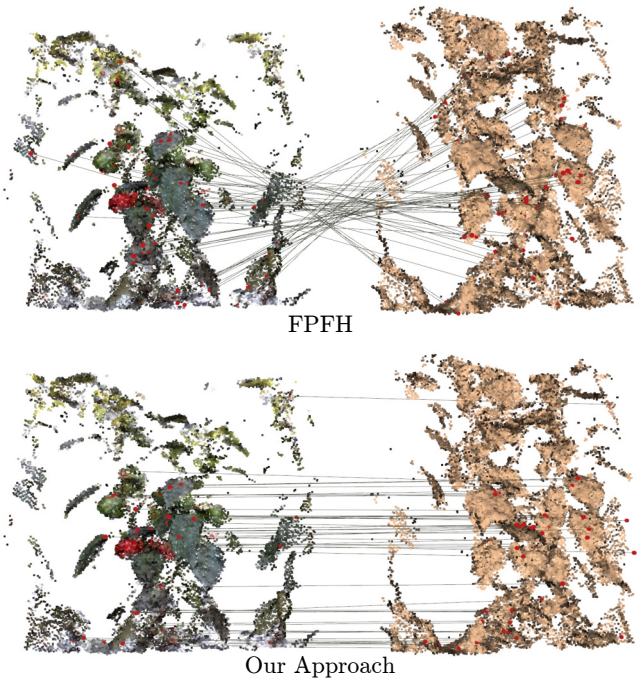

Fig. 5: Top: associations using FPFH 3D descriptors [26]. Bottom: associations obtained with our approach.

To support the claim that our data association mechanism is robust to noise and heavy changes, in Fig. 5 we show how 3D descriptors fail to find associations with our data. They are not able to capture geometric information in this challenging scenario. Our data association approach, instead, can correctly associate points between the two representations.

V. Conclusion

This paper presents a pipeline for 4D mapping of sweet pepper plants in a glasshouse undergoing growth and deformations. We rely only on consumer RGB-D streams to facilitate the adoption of such methods on real platforms. For this reason, our focus is on the ability to process noisy data and incomplete scenes. Furthermore, because plants undergo changes in shape during time, we propose a method that is also robust to large deformations. Our focus is opposed to other methods in the literature based on high-resolution point clouds with complete plant shapes. Our proposed pipeline allows us to re-use a representation of the environment and update it based on new observations. For this reason, it allows tracking the evolution of plants' organs over time. First, we solve odometry using RGB-D images with a frame-to-map algorithm. To be robust to heavy shape changes, we exploit stable plant features. Then, we perform keypoint matching on image pairs between the current stream and the one used to produce the first representation. After that, we translate image matches into 3D point matching. Finally, we exploit those matches to apply a non-rigid deformation

algorithm and update the previous map. A non-rigid ICP method is then applied to refine the result. The evaluation of our system suggests that our pipeline can interpolate plants' organs over time, even in the presence of substantial changes and sensor noise. This can simplify phenotypic trait measurements and the maintenance of plants' digital twins. We see our approach as an essential step towards 4D reconstruction for real agricultural robots.

REFERENCES

[1] A. Afzal and M. Bell. Chapter 11 - Precision agriculture: making agriculture sustainable. In Q. Zaman, editor, *Precision Agriculture*, pages 187–210. Academic Press, 2023.

[2] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 06 2007.

[3] N. Chebrolu, T. Läbe, and C. Stachniss. Spatio-Temporal Non-Rigid Registration of 3D Point Clouds of Plants. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2020.

[4] J. Chen, S. Izadi, and A.W. Fitzgibbon. KinÊtre: animating the world with the human body. *ACM SIGGRAPH*, 2012.

[5] J. Demby's, A. Shafiekhani, F.B. Fritschi, and G.N. DeSouza. Spatio-Temporal Reconstruction and Visualization of Plant Growth for Phenotyping. In *Proc. of the IEEE Symp. Series on Computational Intelligence (SSCI)*, pages 1–8, 2021.

[6] B. Deng, Y. Yao, R.M. Dyke, and J. Zhang. A Survey of Non-Rigid 3D Registration. *Computer Graphics Forum*, 41, 2022.

[7] 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.

[8] H. Ding, B. Zhang, J. Zhou, Y. Yan, G. Tian, and B. Gu. Recent developments and applications of simultaneous localization and mapping in agriculture. *Journal of Field Robotics (JFR)*, 39(6):956–983, 2022.

[9] D. Feil-Seifer, K.S. Haring, S. Rossi, A.R. Wagner, and T. Williams. Where to Next? The Impact of COVID-19 on Human-Robot Interaction Research. *ACM Trans. on Humam-Robot Interaction*, 10(1):7, 2020.

[10] M. Fischler and R. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[11] P. Glira, C. Weidinger, J. Otepka-Schremmer, C. Ressl, N. Pfeifer, and M. Haberler-Weber. Nonrigid Point Cloud Registration Using Piecewise Tricubic Polynomials as Transformation Model. *Remote Sensing*, 15(22), 2023.

[12] X. Gu, Y. Wang, C. Wu, Y. Lee, and P. Wang. HPLFlowNet: Hierarchical Permutohedral Lattice FlowNet for Scene Flow Estimation on Large-Scale Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[14] 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)*, pages 9587–9593, 2023.

[15] M. Innmann, M. Zollhöfer, M. Nießner, C. Theobalt, and M. Stamminger. VolumeDeform: Real-time Volumetric Non-rigid Reconstruction. In *Proc. of the Europ. Conf. on Computer Vision (ECCV)*, 2016.

[16] R. Islam, H. Habibullah, and M. Hossain. AGRI-SLAM: a real-time stereo visual SLAM for agricultural environment. *Autonomous Robots*, 47:1–20, 07 2023.

[17] X. Liu, C.R. Qi, and L.J. Guibas. FlowNet3D: Learning Scene Flow in 3D Point Clouds. In *Proc. of the IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[18] 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.

[19] 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.

[20] R. Newcombe, D. Fox, and S. Seitz. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[21] 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.

[22] S. Paulus, H. Schumann, H. Kuhlmann, and J. Léon. High-precision laser scanning system for capturing 3D plant architecture and analysing growth of cereal plants. *Biosystems Engineering*, 121:1–11, 2014.

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

[24] T. Pire, M. Mujica, J. Civera, and E. Kofman. The Rosario Dataset: Multisensor Data for Localization and Mapping in Agricultural Environments. *arXiv preprint*, arXiv:1809.06413v2, 2018.

[25] R.B. Rusu and S. Cousins. 3d is here: Point cloud library (pcl). In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2011.

[26] R. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2009.

[27] Y. Sahillioğlu. Recent advances in shape correspondence. *The Visual Computer (VC)*, 36:1705 – 1721, 2019.

[28] S. Salti, F. Tombari, and L. Di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Journal of Computer Vision and Image Understanding (CVIU)*, 125:251–264, 2014.

[29] D. Schunck, F. Magistri, R. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. Kuhlmann, and L. Klingbeil. Pheno4D: A spatio-temporal dataset of maize and tomato plant point clouds for phenotyping and advanced plant analysis . *PLOS ONE*, 16(8):1–18, 2021.

[30] A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *Proc. of Robotics: Science and Systems (RSS)*, 2009.

[31] 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 *Proc. of the IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 1761–1771, Jan 2021.

[32] 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.

[33] B. Steder, R. Rusu, K. Konolige, and W. Burgard. NARF: 3D range image features for object recognition. In *Workshop on Defining and Solving Realistic Perception Problems in Personal Robotics at the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010.

[34] R.W. Sumner, J. Schmid, and M. Pauly. Embedded deformation for shape manipulation. *ACM Trans. on Graphics (TOG)*, 26(3):80, 2007.

[35] G. Tam, R. Martin, P. Rosin, and Y.K. Lai. Diffusion Pruning for Rapidly and Robustly Selecting Global Correspondences Using Local Isometry. *ACM Trans. on Graphics (TOG)*, 33, 02 2014.

[36] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, 2005.

[37] 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.

[38] O. Vysotska and C. Stachniss. Lazy Data Association For Image Sequences Matching Under Substantial Appearance Changes. *IEEE Robotics and Automation Letters (RA-L)*, 1(1):213–220, 2016.

[39] M. Wakchaure, B. Patle, and A. Mahindrakar. Application of AI techniques and robotics in agriculture: A review. *Artificial Intelligence in the Life Sciences*, 3:100057, 2023.

[40] 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)*, pages 469–478, 2022.

[41] K. Zampogiannis, C. Fermuller, and Y. Aloimonos. cilantro: A Lean, Versatile, and Efficient Library for Point Cloud Data Processing. In *Proc. of the ACM Intl. Conf. on Multimedia*, MM '18, pages 1364–1367, New York, NY, USA, 2018.

[42] K. Zampogiannis, C. Fermüller, and Y. Aloimonos. Topology-Aware Non-Rigid Point Cloud Registration. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(3):1056–1069, 2021.

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 warrenty 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) *Tiziano Guadagnino* 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/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 iros2024`