



HAL
open science

Multi-scale Space-time Registration of Growing Plants

Haolin Pan, Franck Hétroy-Wheeler, Julie Charlaix, David Colliaux

► **To cite this version:**

Haolin Pan, Franck Hétroy-Wheeler, Julie Charlaix, David Colliaux. Multi-scale Space-time Registration of Growing Plants. The 9th International Conference on 3D Vision (3DV), Dec 2021, Londres, United Kingdom. hal-03402870

HAL Id: hal-03402870

<https://hal.science/hal-03402870v1>

Submitted on 19 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Multi-scale Space-time Registration of Growing Plants

Haolin Pan

Shanghai Jiaotong University
Sony CSL Paris

haolin.pan@polytechnique.edu

Franck Hétroy-Wheeler

University of Strasbourg
hetroywheeler@unistra.fr

Julie Charlaix

ENS Lyon
ju.charlaix@gmail.com

David Colliaux

Sony CSL Paris
koddada@gmail.com

Abstract

In this paper, we introduce a new method for the space-time registration of a growing plant that is based on matching the plant at different geometric scales. The proposed method starts with the creation of a topological skeleton of the plant at each time step. This skeleton is then used to segment the plant into parts that we call branches. Then these branches are further divided into smaller segments that possess a simple geometric structure. These segments are matched between two time steps using a random forest classifier based on their topological and geometric features. Then, for each pair of segments matched, a point-wise registration is devised using a non-rigid registration method based on a local ICP.

We applied our method to various types of plants, including arabidopsis, tomato plant and maize. We established three different metrics for 3D point-wise shape correspondence to test the accuracy, continuity, and cycle consistency of the mapping. We then compared our method with the state-of-the-art. Our results show that our approach achieves better or similar results with a shorter running time.

1. Introduction

In today’s world, an important task for a wide range of real-world agricultural applications, notably plant phenotyping, is the monitoring of plant growth. The manual measurement of a plant is not only tedious, it is also either destructive, thus preventing growth tracking, or inaccurate. Thanks to the development of 3D scanning and photogrammetry technologies [14], we are able to get 3D point clouds from real plants with high resolution. With these acquisitions, we hope to achieve accurate measurements of different organs of the plant (lengths, angles, etc.) Some frameworks have already been proposed to analyze the growth of plants using for example semantic segmentation [21] or skeleton extraction [9], in order to get some global measurements on the growing plant.

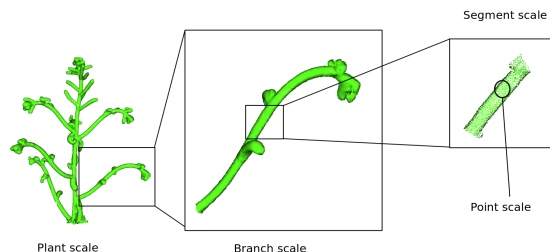


Figure 1: Our registration framework is done at three different scales: branch, segment and then point. We segment the plant into **branches**, then each branch into **segments** which are defined as the largest point subsets between two branching points¹ in the plant.

Our aim is to register point to point the successive 3D point clouds of the plant acquired during its growth. In other words, for each point in one point cloud we need to find its corresponding point in the next point cloud, and find out the point-to-point transformation. Having estimated these correspondences and transformations, we can achieve not only global but also local geometric measurements of the plant. For example, by tracking the points belonging to one specific organ (a branch or a leaf), we can measure how its length or its surface area evolves over time. Moreover, identifying the points that are not mapped from the previous point cloud allows to locate the newly grown organs.

Previous work on 3D shape correspondence and registration has already achieved great results on shapes with a limited number of repetitive structures that move isometrically, such as humans or animals. In these cases local geometric features are adequate to distinguish different positions, with a limited number of landmarks to remove the symmetric ambiguities. These algorithms have achieved great accuracy on data sets with an established ground truth mapping [5], see [38] for a review. However, plants have multiple branches and leaves with similar geometric features and

¹A branching point is defined in the graph sense: it is a graph node with a degree > 2 .

they continuously grow new ones. As we will see, these properties limit the accuracy and performance of traditional algorithms which mostly rely on local geometric features.

A recent approach, dedicated to plants, has proposed to use a skeletonization process to overcome these limits [10]. The overall strategy is to create a skeleton from each point cloud, then establish a correspondence between the skeletons, and finally retrieve the correspondence between points using the skeleton one. This method has satisfying performance on plants with few leaves and branches, such as tomato seedlings. However, the proposed skeleton matching method has limited efficiency and accuracy on plants with a complex structure, such as arabidopsis. Chebrolu et al. [10] use a Hidden Markov Model (HMM) to align the skeletons. The time and space complexity of their method is $O(N^4)$ with N the number of skeleton nodes. Therefore the number of nodes in the skeleton has to be limited in order to have a reasonable computing time and memory usage.

In this paper, we propose a new framework for the space-time registration of growing plants. Unlike the previous methods, we do not establish the point-to-point mapping on the whole plant directly, but consider the shape correspondence at three different scales: branch scale, segment scale and point scale. Figure 1 illustrates the relationship between these scales. Some sample data and a code demo are given in the supplementary material ².

2. Related Work

2.1. Shape correspondence and registration

3D shape registration 3D shape registration is a problem that has been extensively covered. We refer the reader to surveys such as [38, 33, 7] for a general overview, and only mention here methods related to our work.

One of the most widely applied registration methods is the iterative closest point (ICP) algorithm [4], which aims to compute the optimal rigid transformation between two shapes by maximizing the spatial correspondence between them. Building on this, many other methods, e.g. [11], achieve more robust point-wise correspondence and 3D registration by computing handcrafted geometric features, like SHOT [34] or FPFH [32], on 3D point clouds and matching points in feature space instead of the Euclidean space.

Finding a non-rigid transformation is more difficult. Compared with the rigid case, more reliable correspondences and prior assumptions are needed to define the point-wise transformation. Some methods use a predefined skeleton to model the point transformation [40, 2]. Other methods estimate a local rigid transformation for each point or local region to represent a globally non-rigid transformation

[17, 37, 35, 12]. We adopt this strategy in our approach.

3D Shape Correspondence Correspondence between 3D shapes can be achieved by applying the transformation models computed by 3D registration algorithms then searching the nearest neighbours for each transformed point. One of the state-of-art method which does not require an explicit transformation model is the *Functional Maps (FM)* framework originally proposed by Ovsjanikov et al. [27]. This method uses a functional basis (generally a Laplace-Beltrami basis) to represent the shape, so that the mapping between two shapes could be encoded into a matrix with a reasonable dimension. The original method has been extended and improved by much subsequent work, including [18, 13, 26, 30, 20, 22, 28]. Other work has established shape correspondence by learning the features. For example, authors of [6, 24] learn features using a triangle-mesh-based convolutional neural network architecture.

In our framework, we achieve the shape correspondence by estimating a point-wise registration. We compare our results with the *FM*-based methods.

Space-time Registration Space-time registration of a deforming object is a specific type of shape registration. Many solutions [1, 29] rely on prior shape templates and try to deform the templates to fit the target shapes. However, for a growing plant, a shape template is difficult to build since the registration is to be found between two non-isometric shapes. Some other methods use a topological skeleton to establish the correspondence [40, 10]. This seems particularly adapted to the case of plants because of their shape. Some methods dedicated to plants have been proposed in recent years [15, 10]. They rely on plant specific properties, for example a semantic segmentation, to first establish shape correspondence then compute an optimal transformation. We also use a topological skeleton in our approach and compare the results with [15, 10].

3. Methodology

We formulate the 3D point cloud correspondence and registration problem into several different scales: plant, branch, segment and point, as illustrated in Figure 2. At the plant scale, we segment each point cloud into what we call branches and segments. This is done using a topological skeleton. For each of the other three scales, we have designed a specific approach to establish the correspondence between consecutive subsets of the point clouds at this scale. By applying these algorithms hierarchically, we can achieve a point-to-point correspondence with high accuracy.

²Our code is also available at: https://github.com/romi/4d_plant_analysis and the data at: https://zenodo.org/record/5205562#.YVsD_9ozaUk

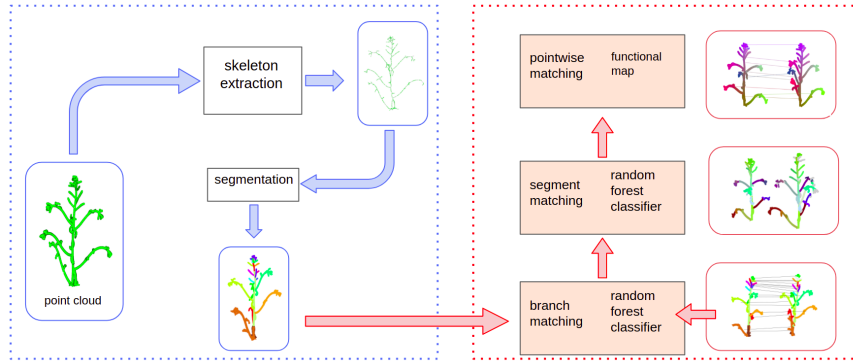


Figure 2: General pipeline of our framework.

3.1. Plant scale: skeleton extraction + segmentation

Given two point clouds of the input plant at two different time steps, our registration framework begins by applying a rigid transformation, estimated by the ICP algorithm [4] in order to align them globally. Our goal here is to guarantee that corresponding plant organs are spatially close and share similar orientations.

Skeleton extraction In our framework, the topological skeleton of the plant plays a significant role and must be as accurate and robust as possible. Any skeleton extraction method can be used, but we found in our experiments that applying iteratively two classical skeleton extraction approaches, namely the *Laplacian Contraction* [8] and the *L1-medial Contraction* [16] methods, allows us to reach satisfying results.

Branch extraction We use the skeleton to segment the plant into parts that we call *branches*, see Fig. 3. One peculiar branch corresponds to the main stem of the plant and is extracted by searching for a path from the lowest node to the highest in the skeleton. Other branches are specified by the collection of skeleton nodes sharing the same topologically closest node on the main stem. The segmentation of the point cloud into branches is derived by assigning each point to its closest skeleton node.

Segment extraction Each branch of the plant point cloud is then split into smaller parts called *segments*, that are again defined thanks to the skeleton. The subset of the skeleton corresponding to the branch can be split into sets of successive nodes and edges separated by a branching node. Points corresponding to such a set form a segment. We define the *base* as the segment of the main stem with the lowest average height.

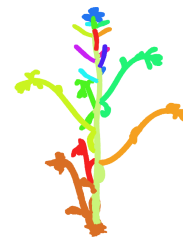


Figure 3: The main stem and branches of an arabidopsis plant. The main stem (light yellow) is the path from the base to the top of the plant. The other branches (red to purple) are sets of segments sharing the same closest point on the main stem.

3.2. Registration at branch and segment scales

Our main idea to register branches and segments between consecutive time steps is to extract a set of features from the target pair of branches or segments. Two random forest classifiers have then been trained to predict the correspondence probability. Finally, we propose a post-processing procedure to handle multiple matching.

3.2.1 Features selection

At the branch scale, the correspondences between main stems are somewhat obvious, since what we have called a main stem refers to the actual main stem of the plant, which is unique for all plants. We also consider that new branches always appear at the top of the plant, which is the case of new organs in the case of the growing plants we have used in our experiments. Therefore, in order to find the correspondence between branches, the height order and angle respect to the main stem are useful features. In our experiments we have also added other geometric features, such as the 3D coordinates of the branch centre.

At the segment scale we have also selected a set of geometric features suited for most kinds of plants. In our framework, a branch could be considered as a graph with its segments as nodes. Most of our features are designed using the properties of this graph, such as its degree. Some other features are also computed using the skeleton, for example, length of the segment.

Tables describing all features used at the branch and the segment scales are given in the supplementary material.

3.2.2 Data generation and augmentation

In order to generate the data set used to learn the correspondences at the branch scale, we manually labeled the correspondence between branches of 7 pairs of consecutive scans from an Arabidopsis plant. We included all 92 pairs of corresponding branches and sampled around 100 pairs of non-corresponding branches. However, using only 200 pairs of branches is very likely to lead to over-fitting. Therefore, we augmented the data set using the following procedures.

Firstly, we have added small perturbations to the point cloud by removing or adding a small number of points. We have also applied a small translation to the branches. This has been done carefully so that the level of the perturbation does not break the correspondence.

We have also simulated acquisition noise. For example, some branches could be lost during the acquisition. In this situation, the height order of the branches above will be affected, and the model fails if it relies too much on the height order correspondence. To increase the robustness of our models against these situations, we intentionally reproduce such acquisition failures in our data augmentation procedure.

We have applied similar procedures to learn correspondences at the segment scale. We have first labeled around 900 pairs of segments with close positions, and then have augmented the data set by adding perturbation and noise.

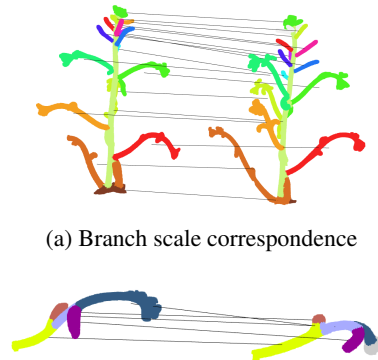
3.2.3 Training and result analysis

Given a pair of branches or segments, we have computed the normalized differences between their features and given them as input to machine learning models. Ideally, the smaller differences, the more likely these two branches or segments are matched. We have split the data set generated as explained in the previous section into a training set and a test set, have trained three classical machine learning models (Random Forest, Logistic Regression and Support Vector Machine) on the train set and have tested their accuracy and F1-score on the test set.

According to these two metrics, the *random forest classifier* is the most efficient of the three models, see Table 1. We thus have selected it as the final model for both scales.

Algorithm	Accuracy	F1-score
Random Forest	0.95 / 0.97	0.91 / 0.96
Logistic Regression	0.95 / 0.64	0.86 / 0.45
SVM	0.92 / 0.87	0.86 / 0.85

Table 1: Accuracy and F1-score of machine learning models for branch or segment matching on the test set. In each case, the score for the branch scale is on the left and the score for the segment scale is on the right.



(a) Branch scale correspondence
(b) Segment scale correspondence between two branches.

Figure 4: Results of branch and segment scale registration. Corresponding branches or segments share the same colour and are connected by lines.

3.2.4 Handling multiple matching

To achieve a complete registration between branches or segments, estimating the probability of correspondence is not enough. We propose a technique to avoid multiple matching, in order each branch or segment in the first point cloud to match a different branch or segment in the second one.

At the branch scale such a multiple matching is uncommon, since different branches are usually far from each other. However it may occur, and in this case we keep the pair with the closest centre position.

Hungarian algorithm On the contrary, multiple matching is very likely to happen at the segment scale if each segment is independently assigned to its most probable match. To solve this issue, we decided to add some constraints and formulate the problem as a linear programming problem.

Given two branches that have been matched, the random forest model trained as explained above has been used to compute the probability of correspondence between each pair of segments of the two branches. Let M be the probability matrix of correspondence: $M[i, j] = P$ means that the probability that the i^{th} segment of the second branch

and the j^{th} segment of the first branch are matched is P .

The correspondence problem boils down to the problem of finding a matrix of selection X with the same dimension as M and binary values: $X[i, j] = 1$ if the i^{th} segment of the second branch can be matched with the j^{th} segment of the first branch. We added two constraints on X . Without loss of generality, let us suppose the first branch has less points than the second branch. The two constraints are that the sum of each row of X should be greater than or equal to 1, and that the sum of each column should be exactly 1. The second constraint guarantees that the correspondences for the second branch are not repetitive, and the first one ensures that the correspondences for the first branch are complete. These constraints could be formulated into linear forms using the masks of columns and rows. The goal of our problem is now to maximize the element-wise production of X and M .

We solve this linear programming problem using the classical *Hungarian algorithm*. Given the solution matrix X , if there are multiple ones in a given row, we select the one with the highest value in M . This way we guarantee that the matching is not repetitive.

Examples of branch/segment scale matching result are illustrated in Figure 4.

3.3. Registration at point scale

After establishing correspondences at the segment scale, our goal is to find the point-wise correspondence between two segments that have been matched. The part of the skeleton corresponding to a segment can be considered as an alternating sequence of nodes and edges with no branching point, thanks to the definition of a segment (see Section 3.1). We call it a *skeleton chain*. Therefore, a segment has a simple geometric structure, for example a cylinder or a rectangle, as illustrated in Figure 1.

To achieve point-wise non rigid registration, we establish local rigid registrations and then smooth them along the skeleton chains. We first divide the skeleton chains of the two segments into the same number of fragments, all with equal lengths. Then for each fragment the transformation can be approximated as a rigid one, as illustrated in Figure 5. For each pair of fragments matched, we compute a rigid transformation T_i using the ICP algorithm. Then for each point p in the point cloud, we obtain the transformed point \hat{p} as a weighted sum of rigid transformations corresponding to the two nearest nodes in the skeleton chain:

$$\hat{p} = \frac{1}{\sum_{i \in NN(p)} \alpha_i} \sum_{i \in NN(p)} \alpha_i * (T_i.p) \quad (1)$$

where $NN(p)$ contains the indices of the two nearest skeleton nodes to p , noting S_i, S_{i+1} . Having p' the projec-

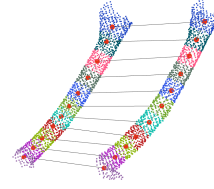


Figure 5: A segment is split into fragments (shown in different colours) where the transformation can be approximated as rigid.

tion of p on edge $S_i S_{i+1}$, then α_i is computed by:

$$\alpha_i = 1 - \frac{\|p' - S_i\|}{\|S_i - S_{i+1}\|}$$

Finally, we retrieve the correspondence by finding the closest point to \hat{p} . Figure 6 shows the result of point-wise correspondence at different scales: segment, branch and whole plant

4. Evaluation

4.1. Metrics

Obtaining a ground truth for point-wise correspondence is difficult therefore we have designed three different metrics to evaluate a point-to-point mapping between two shapes. These metrics quantify the mapping accuracy, its continuity and if it is cycle consistent.

In this section, we note $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ the mapping to be evaluated. $(P_i)_{i \in 1, \dots, M}$, $(Q_i)_{i \in 1, \dots, N}$ are the two point clouds, $t > 0$ is a small value which is used as a distance threshold and is set to the average distance between each point in a point cloud and its nearest neighbour.

Landmark accuracy In our experiments, we have sampled key points of (P_i) and matched them to points of (Q_i) . These key points are defined as *landmarks* in the followings and are considered as ground truth. The landmarks are sampled using the skeleton. For each pair of skeleton nodes matched, we pick their closest points in each point cloud as a pair of landmarks. We evaluate the accuracy of a mapping T by checking if the landmarks are correctly mapped.

Let us note L the selected pairs of landmarks. $(i, j) \in L$ iff P_i is mapped to Q_j . To check if a landmark P_i is correctly mapped using T , we test if the distance between the mapped point $T(P_i)$ and the ground truth mapping Q_j with $(i, j) \in L$ is lower than the threshold t . We compute the ratio of landmarks correctly registered as the score for landmark accuracy:

$$S_{lm} = \frac{\sum_{(i,j) \in L} \mathbb{I}(\|Q_j - T(P_i)\| < t)}{\text{Card}(L)} \quad (2)$$

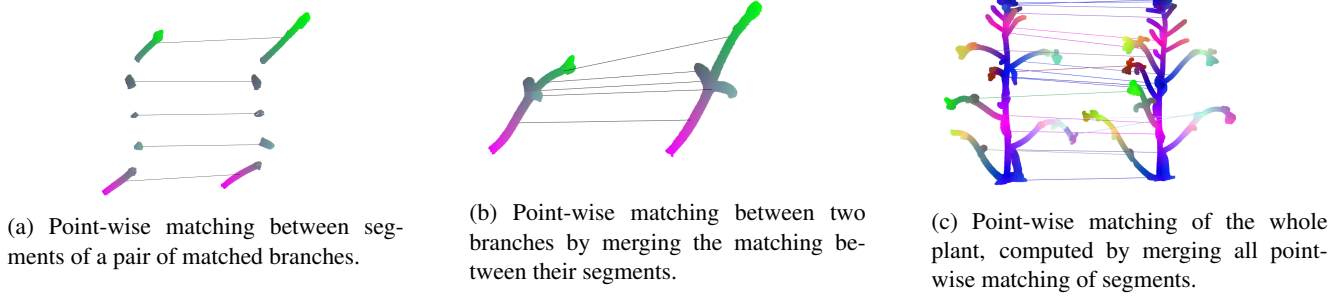


Figure 6: We achieve the shape correspondence by establishing point-wise mapping between shapes. The mapping is visualized by colour: the colour of the target shape (right sides) is determined by the points’ coordinates. The colour of the source shape (left sides) is determined by the correspondence: the correspondent points have the same colour. Several landmarks are randomly sampled and connected to their corresponding points.

Mapping continuity Our second metrics aims to measure the continuity of the mapping on the plant surface. Given a pair of neighbouring points in the point cloud (P_i) (distance between them $\leq t$), we check if their mappings are neighbours in (Q_i).

We only sampled 50% of the points to reduce the computation time. For each point P'_i sampled, we randomly select one of its neighbours P''_i such that $\|P'_i - P''_i\| < t$. Our metrics compute the ratio of pairs of points (P'_i, P''_i) which are mapped to neighbouring points in (Q_i):

$$S_c = \frac{\sum_{i \in 1 \dots M/2} \mathbb{I}(\|T(P'_i) - T(P''_i)\| < t)}{M/2} \quad (3)$$

Cycle consistency Our third metrics has been designed to measure the consistency of the mapping when applied from one point cloud to the next one or the other way round. Let us note T_+ and T_- the point-wise mappings from (P_i) to (Q_i) and from (Q_i) to (P_i), respectively. Ideally, T_- should be the inverse mapping of T_+ . We measure the cycle consistency by checking if each point remains in the neighbourhood of its original position when mapped forth and then back. To speed up the computation, this is done only on half of the points, as for the previous metrics. The cycle consistency metrics is defined as the ratio of points correctly mapped by T_+ then T_- :

$$S_{cc} = \frac{\sum_{i \in 1 \dots M/2} \mathbb{I}(\|T_-(T_+(P'_i)) - P'_i\| < t)}{M/2} \quad (4)$$

4.2. Data sets

We trained the machine learning algorithms on a time sequence of 3D point clouds of arabidopsis. We then applied our framework on sequences of 3D point clouds of plants from various species: arabidopsis (*Arabidopsis*

thaliana), tomato plant (*Solanum lycopersicum*) and maize (*Zea mays*). The arabidopsis point clouds were acquired using a robotic set up where an arm equipped with a camera is moving in circle around the plant, taking 72 images of the plant. The point cloud is then computed using a space carving algorithm. We used the open source hardware and software described in [39]. The 3D data for the two other plant species is open source data³ kindly provided by the authors of [10]. Each point cloud has between 100,000 and 300,000 points. According to our experiments, since tomato and maize sample plants possess relatively simple topological structures, it is rather difficult to identify their main stem. Therefore, we skipped the branch scale registration for them and applied directly the segment scale registration by considering the whole plant as a single branch.

Our experiments were all made using a 8-core *Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz*.

4.3. Visual and quantitative validations

We have selected 3 sample plants of each species (arabidopsis, maize and tomato plant). For each sample plant, we have taken 7 successive point cloud acquisitions with time frame of one day and applied our registration pipeline for each pair of consecutive point clouds. An example of registration results is given in Figure 7 for the tomato plant and the maize cases, where the point-wise correspondence is presented using colours and lines. The complete registration sequence can be found in the supplementary material. We have also tested the robustness of our method on the point clouds with noise or low-resolution. Visual results could be found in the supplementary material.

We have evaluated the results using the three evaluation metrics described in the previous section. Table 2 presents the average registration scores for each sample plant and

³<https://www.ipb.uni-bonn.de/data/4d-plant-registration/>

Sample plant	S_c	S_{lm}	S_{cc}
Tomato plant 1	0.81	0.98	0.91
Tomato plant 2	0.76	0.97	0.84
Tomato plant 3	0.77	0.97	0.85
Average on tomato plant	0.78	0.97	0.86
Maize 1	0.83	0.96	0.91
Maize 2	0.77	0.91	0.79
Maize 3	0.80	0.97	0.82
Average on maize	0.80	0.95	0.84
Arabidopsis 1	0.78	0.78	0.79
Arabidopsis 2	0.80	0.78	0.81
Arabidopsis 3	0.75	0.92	0.83
Average on arabidopsis	0.77	0.88	0.80

Table 2: Average scores for the registration of sample plants: S_c is the score for continuity, S_{lm} for landmark accuracy and S_{cc} for cycle consistency.



Figure 7: Result of our framework on one tomato plant (left) and one maize (right): corresponding points share the same colour. Some landmarks have been sampled and are shown connected to their corresponding points.

each metrics. We can observe that our framework has better performance with tomato plant and maize than with arabidopsis, even though the random forest classifier for the segment scale matching was trained on arabidopsis. One reason is that the tomato plant and maize sample plants have simpler geometric and topological structures, with fewer branches and leaves, making the registration task simpler. This also shows that our features selected for the random forest classifiers can be generalized to many plants.

4.4. Ablation study on arabidopsis

Our pipeline relies on registrations at different scales. For plants with a complex structure like arabidopsis, all scales are necessary to reach the optimal result. To illustrate this, we have designed two experiments showing the respective contribution of each core component of our pipeline.

Our first experiment was to skip the branch scale and segment scale registrations, directly using the skeleton matching algorithm as well as the point transformation algorithm presented in [10] in order to establish a point-wise mapping. Here, to fully cover the geometric structure of the

Registration framework	S_c	S_{lm}	S_{cc}	time
Our pipeline	0.78	0.79	0.79	300 s
HMM skeleton matching + point transformation	0.35	0.41	0.15	≥ 2 days
Branch & segment matching + FM	0.55	0.91	0.41	8 hours

Table 3: Average scores and running time for the registration of 7 acquisitions of a sample arabidopsis using three different frameworks. The running time for the skeleton extraction and the segmentation is not included.

arabidopsis plant but also to speed up the running time, we down-sampled the skeleton to around 60 nodes. However, due to the high temporal and spatial complexity of the point transformation algorithm ($O(N^4)$), the registration needed 11 Gb of memory and 2 days to be computed. Moreover, the accuracy of skeleton matching for the parts of the plant far from the base revealed to be far worse than for the ones next to it. This is because of the uncertainty accumulation when the HMM is used. Consequently, only 30% of skeleton nodes were correctly matched.

The second experiment was to replace our point-wise scale registration, described in Section 3.3, by a state-of-the-art non-rigid 3D shape correspondence algorithm, namely *FM* [27], which establishes the point-wise correspondence by computing local geometric descriptors. We have used an open-source code⁴ that implements not only the basic *FM* framework but also many improvements [25, 31, 18, 26, 13, 30, 19, 23, 20]. The geometric descriptors used are the Heat Kernel Signature [36] and the Wave Kernel Signature [3].

We have conducted these two experiments on 7 acquisitions of one sample arabidopsis plant and evaluated the registration results using our three metrics. Results are shown in Table 3. Note that the landmarks used to compute the landmark accuracy score are same for the two experiments as with our pipeline.

We can observe that the HMM-based skeleton matching algorithm is less accurate and far more time consuming for arabidopsis. Our pipeline achieves better continuity and cycle consistency scores than *FM*-based pipeline with less running time.

4.5. Comparison to the related work

We have also compared our approach with several reference methods on the *tomato* and *maize* plants. First, we have applied the *HMM*-based method [10], designed for

⁴<https://github.com/RobinMagnet/pyFM>

Method	S_c	S_{lm}	S_{cc}	time
Our pipeline	0.81 / 0.80	0.98 / 0.97	0.89 / 0.82	20 s
HMM-based [10]	0.56 / 0.44	0.93 / 0.90	0.52 / 0.15	100 s
Segmentation + FM	0.54 / 0.55	0.90 / 0.88	0.41 / 0.43	2 hours
FM (on whole plant)	0.39 / 0.28	0.55 / 0.38	0.28 / 0.18	4 hours

Table 4: Average scores and running time for the registration of 7 acquisitions of one tomato plant or maize (scores for the tomato plant in black and for the maize in red), using different registration frameworks. Note that the running time for the pre-processing stage at plant scale (skeleton extraction and segmentation) is not included for all methods.

these data sets, and evaluated their results with our evaluation metrics. We have also compared our work to the *FM* approach, first by applying the *FM* framework directly on the whole point cloud and second by segmenting the plant and matching its segments, then applying *FM* between each pair of segments. The scores and running time of these experiments are presented in Table 4. For all three metrics we consider, we observe that our pipeline outperforms the other approaches on the plant point clouds provided by [10]. The *FM*-based methods take a longer time than the other methods because the computation of the Laplace-Beltrami basis and of the geometric signatures is time consuming. We have also tested the Smooth Shells method [12] on the arabidopsis data. This method gave less satisfying results than on human meshes with $S_c = 0.69$ and $S_{lm} = 0.48$, and is very time-consuming as other *FM*-based methods: it took 1 hour for one pair of shapes in our case. Additionally, it requires high-quality meshes as input, which may be difficult to obtain from plant point clouds.

5. Applications

Our point-wise registration framework can be applied to a variety of problems. We now describe two examples.

Temporal interpolation between scans The ability to interpolate between temporal acquisitions is useful for analyzing plants even when some actual acquisitions are not available. It allows us to predict both the motion and growth of the plant at any time instant. We can for example apply a simple linear interpolation between point clouds. Let P_1 and P_2 be the point clouds of a sample plant at times t_1 , t_2 . For each pair of matched points $(p_1, p_2) \in (P_1, P_2)$, the interpolated point at time $t \in [t_1, t_2]$ is computed as $\hat{p}_t = \frac{1}{t_2 - t_1}((t - t_1)p_2 + (t_2 - t)p_1)$. Points in P_2 without

corresponding point in P_1 (the new emerging segments) can be attached to the matching points of their nearest neighbour. The set $\{\hat{p}_t\}$ is the interpolated point cloud at t .

Our results using this approach on a maize plant can be visually compared with the interpolation result of [15] visually, see the supplementary material. We have not carried a quantitative comparison since it is hard to design a proper evaluation metrics for the interpolation problem. Our framework achieves visually similar results to [15] with similar running time. A series of videos showing the interpolated point clouds with our approach can also be found in the supplementary material.

Quantitative growth tracking Our framework can also be used to compute phenotypic traits on plants and track them temporally. For example, by computing the length of the skeleton we can estimate the length of the main stem. We have manually measured the growth of the main stem of an arabidopsis and compared to measurements given by our approach. It shows that the error made by our approach does not exceed 5%. We can also track the length of different branches over time, as shown in the supplementary material.

6. Conclusion

In this paper we have proposed a new 3D space-time registration framework for plants based on establishing correspondences at different scales. Our framework is applicable to various species of plants and has been tested on arabidopsis, tomato plant and maize. We have also proposed three metrics to evaluate 3D shape mappings and used them to compare our approach with the state-of-art. Our experiments have shown that our method achieves better or similar results with a shorter running time.

One of the main limitations of the proposed registration framework is that a clean topological skeleton of the plant is needed. This may be difficult to obtain in outdoor environments because of acquisition noise, occlusions, etc. Moreover, computing a skeleton can be difficult or even ambiguous for plants with broad leaves. We have not tested our approach on such scenarios but we expect that more work is yet to be done to clean the input point clouds and improve the skeleton extraction algorithms in such cases.

Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 773875. The authors would like to thank Christophe Godin (Inria) and Fabrice Besnard (CNRS/ENS Lyon) for their valuable comments.

References

- [1] Naveed Ahmed, Christian Theobalt, Petar Dobrev, Hans-Peter Seidel, and Sebastian Thrun. Robust fusion of dynamic shape and normal capture for high-quality reconstruction of time-varying geometry. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008. **2**
- [2] Brett Allen, Brian Curless, and Zoran Popović. Articulated body deformation from range scan data. *ACM Transactions on Graphics*, 21(3):612–619, July 2002. **2**
- [3] Mathieu Aubry, Ulrich Schlickewei, and Daniel Cremers. The wave kernel signature: A quantum mechanical approach to shape analysis. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1626–1633, 11 2011. **7**
- [4] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. **2, 3**
- [5] Federica Bogo, Javier Romero, Matthew Loper, and Michael J. Black. Faust: Dataset and evaluation for 3d mesh registration. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3794–3801, 2014. **1**
- [6] D. Boscaini, Jonathan Masci, E. Rodolà, and M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *NIPS*, 2016. **2**
- [7] Sofien Bouaziz, Andrea Tagliasacchi, Hao Li, and Mark Pauly. Modern techniques and applications for real-time non-rigid registration. In *SIGGRAPH ASIA 2016 Courses*, SA '16, New York, NY, USA, 2016. Association for Computing Machinery. **2**
- [8] Junjie Cao, Andrea Tagliasacchi, Matt Olson, Hao Zhang, and Zhixun Su. Point cloud skeletons via laplacian based contraction. In *SMI 2010 - International Conference on Shape Modeling and Applications, Proceedings*, pages 187–197, 06 2010. **3**
- [9] Ayan Chaudhury and Christophe Godin. Skeletonization of Plant Point Cloud Data Using Stochastic Optimization Framework. *Frontiers in Plant Science*, 11, June 2020. **1**
- [10] Nived Chebrolu, Federico Magistri, Thomas Läbe, and Cyrill Stachniss. Registration of spatio-temporal point clouds of plants for phenotyping. *PloS one*, 16(2):e0247243, 2021. **2, 6, 7, 8**
- [11] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 06 2015. **2**
- [12] M. Eisenberger, Z. Löhner, and D. Cremers. Smooth shells: Multi-scale shape registration with functional maps. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, 2020. **2, 8**
- [13] Danielle Ezuz and Mirela Ben-Chen. Deblurring and denoising of maps between shapes. *Computer Graphics Forum*, 36:165–174, 08 2017. **2, 7**
- [14] Jonathan A. Gibbs, Michael Pound, Andrew P. French, Darren M. Wells, Erik Murchie, and Tony Pridmore. Approaches to three-dimensional reconstruction of plant shoot topology and geometry. *Functional Plant Biology*, 44:62–75, 2016. **1**
- [15] Tim Golla, Tom Kneiphof, Heiner Kuhlmann, Michael Weinmann, and Reinhard Klein. Temporal upsampling of point cloud sequences by optimal transport for plant growth visualization. *Computer Graphics Forum*, 39, 05 2020. **2, 8**
- [16] Hui Huang, Shihao Wu, Daniel Cohen-Or, Minglun Gong, Guiqing Li, and Baoquan Chen. L-1-medial skeleton of point cloud. *ACM Transactions on Graphics*, 32, 07 2013. **3**
- [17] Qixing Huang, Bart Adams, Martin Wicke, and Leonidas Guibas. Non-rigid registration under isometric deformations. *Computer Graphics Forum*, 27:1449–1457, 07 2008. **2**
- [18] Qixing Huang, Fan Wang, and Leonidas Guibas. Functional map networks for analyzing and exploring large shape collections. *ACM Transactions on Graphics*, 33(4), July 2014. **2, 7**
- [19] Ruqi Huang, Panos Achlioptas, Leonidas Guibas, and Maks Ovsjanikov. Limit shapes – a tool for understanding shape differences and variability in 3d model collections. *Computer Graphics Forum*, 38:187–202, 08 2019. **7**
- [20] Ruqi Huang, Jing Ren, Peter Wonka, and Maks Ovsjanikov. Consistent zoomout: Efficient spectral map synchronization. *Computer Graphics Forum*, 39(5):265–278, 2020. **2, 7**
- [21] Yangyan Li, Xiaochen Fan, Niloy J. Mitra, Daniel Chamovitz, Daniel Cohen-Or, and Baoquan Chen. Analyzing growing plants from 4d point cloud data. *ACM Transactions on Graphics*, 32(6), Nov. 2013. **1**
- [22] O. Litany, E. Rodolà, A. M. Bronstein, M. M. Bronstein, and D. Cremers. Non-rigid puzzles. In *Proceedings of the Symposium on Geometry Processing*, SGP '16, page 135–143, Goslar, DEU, 2016. Eurographics Association. **2**
- [23] Simone Melzi, Jing Ren, Emanuele Rodolà, Abhishek Sharma, Peter Wonka, and Maks Ovsjanikov. Zoomout: Spectral upsampling for efficient shape correspondence. *ACM Transactions on Graphics*, 38(6), Nov. 2019. **7**
- [24] Federico Monti, D. Boscaini, Jonathan Masci, E. Rodolà, Jan Svoboda, and M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5425–5434, 2017. **2**
- [25] A. Nguyen, M. Ben-Chen, Katarzyna Welnicka, Y. Ye, and L. Guibas. An optimization approach to improving collections of shape maps. *Computer Graphics Forum*, 30, 2011. **7**
- [26] Dorian Nogneng and Maks Ovsjanikov. Informative descriptor preservation via commutativity for shape matching. *Computer Graphics Forum*, 36:259–267, 05 2017. **2, 7**
- [27] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas Guibas. Functional maps: A flexible representation of maps between shapes. *ACM Transactions on Graphics*, 31(4), July 2012. **2, 7**
- [28] Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast Sinkhorn Filters: Using Matrix Scaling for Non-Rigid Shape Correspondence with Functional Maps. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville (virtual), United States, June 2021. **2**
- [29] Yuri Pekelny and Craig Gotsman. Articulated object reconstruction and markerless motion capture from depth video. *Computer Graphics Forum*, 27(2):399–408, 2008. **2**

- [30] Jing Ren, Adrien Poulenard, Peter Wonka, and Maks Ovsjanikov. Continuous and orientation-preserving correspondences via functional maps. *ACM Transactions on Graphics*, 37(6), Dec. 2018. [2](#), [7](#)
- [31] Raif Rustamov, Maks Ovsjanikov, Omri Azencot, Mirela Ben-Chen, Frédéric Chazal, and Leonidas Guibas. Map-based exploration of intrinsic shape differences and variability. *ACM Transactions on Graphics*, 32, 07 2013. [7](#)
- [32] Radu Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *IEEE Int. Conf. Robot. Automation (ICRA)*, pages 3212 – 3217, 06 2009. [2](#)
- [33] Y. Sahillioglu. Recent advances in shape correspondence. *The Visual Computer*, 36:1705 – 1721, 2019. [2](#)
- [34] Samuele Salti, Federico Tombari, and Luigi Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125, 08 2014. [2](#)
- [35] Olga Sorkine and Marc Alexa. As-rigid-as-possible surface modeling. In *Symposium on Geometry Processing*, pages 109–116, 01 2007. [2](#)
- [36] Jian Sun, Maks Ovsjanikov, and Leonidas Guibas. A concise and provably informative multi-scale signature based on heat diffusion. *Computer Graphics Forum*, 28:1383–1392, 07 2009. [7](#)
- [37] J. Süßmuth, M. Winter, and G. Greiner. Reconstructing animated meshes from time-varying point clouds. *Computer Graphics Forum*, 27, 2008. [2](#)
- [38] Gary K.L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, Frank C. Langbein, Yonghuai Liu, David Marshall, Ralph R. Martin, Xian-Fang Sun, and Paul L. Rosin. Registration of 3d point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013. [1](#), [2](#)
- [39] Timothée Wintz, David Colliaux, Peter Hanappe, and Sony CSL Paris. Automated extraction of phyllotactic traits from *arabidopsis thaliana*. In *CVPPP workshop, ECCV 2018*, 2018. [6](#)
- [40] Qian Zheng, Andrei Sharf, Andrea Tagliasacchi, Baoquan Chen, Hao Zhang, Alla Sheffer, and Daniel Cohen-Or. Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum*, 29:635–644, 05 2010. [2](#)