2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

# Spatio-Temporal Correspondence Estimation of Growing Plants by Hausdorff Distance based Skeletonization for Organ Tracking

Sharmistha B Pandey[1] David Colliaux[2] Ayan Chaudhury[1]

***Abstract*— Tracking of plant organs over spatio-temporal sequence of point cloud data is one of the demanding tasks of agricultural robotics for automated plant monitoring and growth analysis. Due to the complex geometry of plants, it is extremely difficult to identify and track the individual organs in different growth stages of plants. In this paper, we present an approach to perform correspondence estimation of different plant organs over a series of spatio-temporal data. The approach is based on two stages. In the first stage we develop a robust skeleton extraction method from unstructured plant point cloud data by adopting Hausdorff distance metric and modified breadth first search algorithm. The proposed skeletonization method is shown to be performing better than state-of-the-art, especially in handling very thin and delicate branches. We also address an overlooked problem of connecting skeleton points in the form of a graph, and demonstrate that different types of plant phenotype parameters can be obtained in a fully automatic manner from the skeleton graph. In the second stage, we exploit the skeleton graphs in developing an algorithm to perform correspondence estimation among the skeleton nodes using a cosine similarity based approach. We demonstrate the effectiveness of the proposed skeletonization technique in tracking different organs of the plant by finding good quality correspondences. Experiments are performed on three datasets on real and synthetic sequence of spatio-temporal plant point cloud data to demonstrate the effectiveness of the proposed method.**

## I. Introduction

In recent years, automated plant phenotyping [1] has gained much attention in the robot vision community, and have witnessed great progress in different application areas [2], [3], [4], [5]. With the advancements of imaging and sensor devices [6], 3D point cloud based analysis of plants have been an integral part of agricultural automation and plant phenotyping [7], [8], [9]. Different types of applications of 3D point cloud based plant phenotyping include organ segmentation, growth analysis of plants, branch pruning, automatic counting of number of branches, etc. Among all these different types of applications, tracking of plant organs have gained much attention recently [10], [11], [12], [13], [14]. Figure 1 shows an example of correspondence estimation in a temporal sequence of a growing plant where the correspondences at each organ level can be used for tracking organs over the time series data.

While organ tracking is a challenging problem itself, we argue that having an accurate skeletonization algorithm can ameliorate the complexity of the problem to a great extent. For example, from a skeleton graph it is very easy to determine the number of branches in the plant, their lengths, angles, etc. However, it is not a trivial task to extract the skeleton graph from the unstructured point cloud data. Depending on the complexity of the plant, the extracted skeleton may not be accurate enough to capture the topology and geometry of the branching details, which may lead to wrong phenotypic parameter estimation.

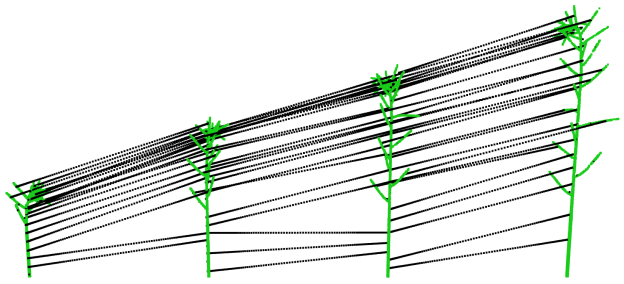

Fig. 1. An example of spatio-temporal correspondence matching of a growing plant by the proposed method. From left to right: day 0, day 6, day 9, day 12 growth stages of the plant. Correspondences are shown in black lines.

Skeleton extraction from 3D point cloud data has been a long standing problem in computer graphics applications [15], [16], [17]. However most of the methods are built for general 3D objects and not specifically for plants. State-of-the-art skeletonization techniques specifically built for plants mostly rely on different types of assumptions on the plant structure. In an earlier work, Xu *et al.* [18] quantized the point cloud into discrete parts and connected the centroid of these parts to form a skeleton graph. Although the method is simple to use and works well for trees with thick and long branches, it fails to extract accurate skeleton for very thin and short branches. In a similar line of work, Bucksch *et al.* [19] subdivided the point cloud into octree cells and estimated skeleton graph from this representation. Skeletonization methods developed for geometric tree reconstruction for computer graphics applications [20], [21] tend to generate skeletons which need not be necessarily biologically relevant to the original plant structures. Optimization based approaches have also been shown to be successful for different application areas as well. Au *et al.* [22] performed geometry contraction of the 3D shape by applying Laplacian smoothing technique. The method assumes polygonal mesh format of the input data. ROSA skeletonization technique [23] performs skeletonization of incomplete point cloud

This work is supported by DST SRG project SRG/2022/000510.
The authors would like to thank Christophe Godin of INRIA for creating the virtual plant models.

[1] Department of CSE, Indian Institute of Technology Kharagpur (bhattacharjee.sharmistha88@gmail.com, ayanc@cse.iitkgp.ac.in)
[2] Sony CSL Paris (koddda@gmail.com)

data by rotational symmetry axis of an oriented point set. However the technique performs well on a handful types of shape varieties. Laplacian based contraction is also adopted by Cao *et al.* [24]. They considered raw point cloud data and exploited the idea of local Delaunay triangulation and topological thinning for skeleton extraction. Later Wu *et al.* [25] used the idea to skeletonize Maize plant point cloud data. $L_1$ median skeleton [26] is also shown to be robust to different types of objects including plant structures.

One of the most prevalent problems with the state-of-the-art algorithms is that, the extracted skeleton for thin and short branches that comprises of very few points, does not capture the actual topology of the original plant point cloud. Also, often the computed skeleton points fall outside the boundary of the point cloud, thus making the skeleton to be biologically irrelevant. Another overlooked problem with skeletonization is the ignorance of graph formation from the skeleton points. A trivial approach to connect the extracted skeleton points by nearest neighbor strategy does not take into consideration the original branching topology, and thus often leads to incorrect connection of the skeleton points, especially for close branches.

Skeletonization has been exploited by some methods for spatio-temporal registration of plant point clouds. Chebrolu *et al.* [14] presented a Hidden Markov Model (HMM) based approach to find efficient matching of skeleton nodes. However it is assumed that a good skeleton is available, and developing the skeleton structure is overlooked in the work. Pan *et al.* [11] also used skeleon in performing multi-scale registration of 4D data. The skeleton is constructed by using Laplacian contraction [24] and L1 median [26], and it is assumed that the skeleton is accurate. Segmentation of plant organs followed by 4D registration is performed by Magistri *et al.* [10]. Fast Feature Point Histogram (FPFH) are used for computing the feature vector, which are then used by Support Vector Machine (SVM) to perform organ segmentation. Registration is performed at each organ level after performing the segmentation. Recently, Heiwolt *et al.* [12] explited statistical shape of the leaves to perform the registration process. Most of the abovementioned works consider the Maize and Tomato plant dataset from Pheno4D [27].

Our work is different from the discussed method in multiple ways. We propose two major contributions in this paper: robust skeleton graph extraction from unstructured plant point cloud data, and a cosine similarity based approach to match the skeleton nodes at the organ level. More specifically, we present a multi-stage approach of skeleton extraction from unstructured 3D plant point cloud data. In the first stage, we exploit the agglomerative clustering on the original point cloud data, and adopt the Hausdorff distance metric for finding the associativity among the neighboring skeleton nodes. In the next stage, a modified Breadth First Search (BFS) algorithm is developed to connect the skeleton nodes to form a graph. Unlike the typical approach of forming the skeleton graph after finding the skeleton points, we connect the skeleton points while forming the clusters, so that the skeleton graph can be obtained from the inherent structure of the original plant point cloud data. From the obtained skeleton graph, we define attributes to every skeleton node, and adopt the cosine similarity matching to perform the matching of skeleton graphs over the temporal sequence. We demonstrate the effectiveness of both skeletonization and registration method on several challenging datasets from real world plants as well as generated synthetic plant datasets.

The rest of the paper is organized as follows. In the next section we describe the core methodology of skeleton extraction in detail. Next we describe the correspondence estimation algorithm, followed by experimental results with both qualitative and quantitative results along with runtime analysis comparison with the state-of-the-art. Finally some future research directions are outlined.

## II. Skeleton Graph Extraction

In this section, we present the details of skeletonization algorithm. The proposed method is fully automatic, and the tunable parameters have been empirically found to have an optimum fixed value for all the different plant structures studied in this paper.

Given a set of unstructured point cloud $\mathcal{X} = \{\mathbf{p_i} \in \mathbb{R}^3 | i = 1, 2, \cdots, m\}$, the goal is to obtain a set of skeleton points $\mathcal{S} = \{\mathbf{s_j} \in \mathbb{R}^3 | j = 1, 2, \cdots, n\}$ as a rooted tree graph $\mathcal{G} = \{V_s, E_s\}$ embedded in $\mathbb{R}^3$, where $V_s = \{\mathbf{s_j}\}_{1 \leq j \leq n}$ is the set of vertices and $E_s = (s_p, s_q)$ is the set of edges connecting the ordered pair of vertices $s_p$ and $s_q$. For simplicity, the root of the skeleton graph is defined as the bottom most point (or the point with the lowest $z$ value, assuming that the $z$-axis is vertical).

Since the plant is scanned using a laser scanner or depth sensor instrument, the generated raw point cloud data may differ drastically in terms of the number of scanned points in accordance to the camera resolution and the tree structure under consideration. A very high scan density may record huge number of points, processing which can be computationally expensive. So first we downsample the number of points to $\eta << m$ for faster processing.

### A. Agglomerative Clustering for Grouping

The objective is to group the point cloud into hierarchical clusters to form the basis of skeleton nodes by taking the median of each cluster. Following a two-stage clustering approach, first we find the coarse clusters and then refine them for very thin branches by repeating the process with the generated smaller clusters. Although an obvious way to obtain the clusters is to quantize the $z$-coordinate into discrete intervals (as in [18], [8]), but this type of grouping may result in wrong skeleton point computation. Tree branches belonging to the same horizontal axis will be grouped as a single cluster irrespective of their actual topologies (as shown in the left column of Fig. 2), which will eventually result in computing skeleton points at the median of two completely different branches.

We perform agglomerative clustering on the point cloud, which is a bottom-up approach that generates localized

Fig. 2. Naive vertical quantization (left column) versus agglomerative clustering (right column). Each color represents one cluster (there are repetition of colors). Note that the number of clusters is the same in both cases, although visually they might look different. The zoom-in shows an example of second level clustering result.

clusters of points based on Euclidean distance among the points rather than globalized clusters for each point from a central point. The inevitable advantage of this type of clustering for plant structure topology is that, the clustering provides associativity among points belonging to the same sub-structure, as well as the number of required clusters ($\tau$) can be controlled (in all of our experiments, we keep $\tau = 35$). In the agglomerative clustering, we minimize the variance of the clusters for being merged into a single cluster. In other words, we compute the distance between clusters as the sum of squared differences within all clusters. Exploiting the fact that very thin branching structures are likely to be present near the canopy of a tree, we conduct the second level clustering on $1/3$-rd of the plant height from the top (this is a tunable parameter). Each cluster will ultimately represent one or more skeleton nodes in the final skeleton graph. In Fig. 2 second column, the first level clusters are shown in different colors, and an example of second level clustering is shown with the zoom-in view.

### B. Associativity by Hausdorff Distance Metric

Since we aim at forming a skeleton graph from the point cloud, the centroids of respective clusters obtained from the agglomerative clustering needs to be connected so that the original branching topology is maintained. Typical distance metrics (e.g. Euclidean, Minkowski, Mahalanobis etc) for finding the closeness among the generated points in 3D structure of plants measures only the physical distance between the points under consideration. We argue that with this type of configuration, two points belonging to separate branches may be labelled to be associated if they are physically near rather than two points which may be on the same branch but at a greater distance. Instead, we adopt Hausdorff distance metric and find the "nearness" of the clusters accounting for each of the point in the clusters in each of the dimesion separately and obtain the connectivities among the skeleton nodes. In general, Hausdorff distance measures the distance between subsets of the same metric space, which is the maximum distance between two sets of points. Given two clusters $\mathcal{C}_i = \{c_1^i, c_2^i, \cdots, c_p^i\}$ and $\mathcal{C}_j = \{c_1^j, c_2^j, \cdots, c_q^j\}$, we compute the bidirectional Hausdorff distance $\delta_H(\cdot)$ between $\mathcal{C}_i$ and $\mathcal{C}_j$ as,

$$\delta_H(\mathcal{C}_i, \mathcal{C}_j) = \max(\tilde{\delta}_H(\mathcal{C}_i, \mathcal{C}_j), \tilde{\delta}_H(\mathcal{C}_j, \mathcal{C}_i)), \tag{1}$$

where $\tilde{\delta}_H(\mathcal{C}_i, \mathcal{C}_j) = \max_{\phi_i \in \mathcal{C}_i} \min_{\phi_j \in \mathcal{C}_j} ||\phi_i - \phi_j||$.

Given all the clusters, we compute $\delta_H(\cdot)$ for all cluster pairs and find the best match among them. At this point we have the nearest pair of clusters among all clusters and name the centroids of such cluster pairs to form the *Hausdorff pairs*.

### C. BFS based Skeleton Graph Formation

With the generated Hausdorff pairs, we construct the skeleton graph using a modified Breadth First Search (BFS) algorithm. Each node of the Hausdorff pairs form the skeleton nodes. The Hausdorff pairs are traversed iteratively from the lowermost node (as source node) in a BFS traversal technique to finally generate a single connected component. In each iteration, the connected components are removed from the Hausdorff pairs and the traversal is repeated with the remaining Hausdorff pairs until all the pairs are exhausted. Based on the previously computed Hausdorff distances, these connected components are interconnected to form a connected skeleton graph. The process is described in Algorithm 1, which takes as input the Hausdorff pairs ($\mathcal{H}$) and the set of all centroids ($\alpha$). The function BFS($\cdot$) is iteratively called until all the Hausdorff pairs are exhausted. In each iteration, the generated connected component is a subset (or proper subset) of the final skeleton graph. The whole skeleton extraction process is described in Algorithm 2.

**Algorithm 1** BFS based graph formation (BFSG)

**Require:** $\{\mathcal{H}\}$, $\{\alpha\}$
**Ensure:** Skeleton graph $G = (\alpha, e)$
  $\mathcal{T} = \varnothing$ ▷ $\mathcal{T}$ is a set of connected components
  **while** $\mathcal{H}$ is not empty **do**
    rootnode = $\min(\{\alpha\})$
    $\mathcal{T}$ = $\mathcal{T}$ + $BFS$(rootnode, $\mathcal{H}$)
    $\mathcal{H}$ = $\mathcal{H}$ − $\mathcal{T}$
    $\{\alpha\} = \{\alpha\} - \{\alpha_{\mathcal{T}}\}$ ▷ $\alpha_{\mathcal{T}}$ is the set of centroids in $\mathcal{T}$
  **end while**
  $\{e\} \leftarrow \delta_H(\mathcal{T}_i, \mathcal{T}_j)_{\forall i,j=1,2,\cdots,|\mathcal{T}| \atop i \neq j}$

## III. Cosine Similarity based Correspondence

Given the skeleton graphs $G_1, G_2, \cdots, G_t$ from the temporal sequence, the goal is to find the pairwise mapping of nodes between $(G_1, G_2), (G_2, G_3), \cdots, (G_{t-1}, G_t)$. Let two skeleton graphs $G_1$ and $G_2$ have *m* and *n* number of nodes respectively. Let $c_{ij}$ denotes the cost of matching of node *i* with node *j* ; $\gamma_{ij} \in \{0, 1\}$ denotes the assignment/correspondence

**Algorithm 2** Skeleton extraction algorithm

**Require:** $\mathcal{X} = (\mathbf{p_1}, \cdots, \mathbf{p_n})^T$ where $\mathbf{p_i}(x_i, y_i, z_i) \in \mathbb{R}^3$, parameter $\tau$
**Ensure:** Skeleton graph $G = (\alpha, e)$ ▷ '$\alpha$' nodes, 'e' edges
$\eta_d \leftarrow$ DownSample($\mathcal{X}$)
$\{\eta_{d_1}, \eta_{d_2}, ..., \eta_{d_\tau}\} \leftarrow$ Aggl($\eta_d, \tau$) ▷ Aggl. clustering
**for** $i = 1$ to $\tau$ clusters **do**
  $\alpha_i \leftarrow \sum_{j=1}^{|\eta_{d_i}|} \mathbf{p_j}/|\eta_{d_i}|$
**end for**
$\{\alpha\} \leftarrow \biguplus_{i=1}^{\tau} \alpha_i$ ▷ $\alpha$ is a set that contains all the centroids
**for** all $i = 1 \cdots \tau$ **do**
  **for** all $j = 1 \cdots \tau$ **do**
    **if** $i \neq j$ **then**
      $\{\zeta_i^j\} \leftarrow \delta_H(\eta_{d_i}, \eta_{d_j})$ ▷ $\zeta_i^j$ is the maximum distance between $\eta_{d_i}$ and $\eta_{d_j}$
    **end if**
    $\{\phi_k^{ij}\} \leftarrow \min(\{\zeta_i^j\})$ ▷ $\phi_k^{ij}$ is the Hausdorff pair between *i*-th and *j*-th cluster, $k$ is initialized to 1
  **end for**
**end for**
$G$ = BFSG($\{\phi_k^{ij}\}, \{\alpha\}$)

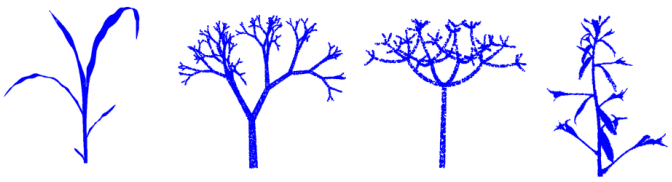

Fig. 3. Sample plant point clouds from the datasets. From left to right: Maize [27], PypeTree [28], Synthetic data, *Arabidopsis* [11].

; and $\mathcal{P}$ denotes the set of $(m \times n)$ permutation matrices. Then the assignment problem can be written as,

$$\min_{\Gamma \in \mathcal{P}} f(\Gamma) = \arg\min_{\gamma} \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}\gamma_{ij} = \arg\min_{\Gamma} \Gamma^T \mathbf{C}, \quad (2)$$

such that,

$$\Gamma \in \mathcal{P}, \mathcal{P} := \Big\{\Gamma | \gamma_{ij} = \{0, 1\}, \sum_{j=1}^{n} \gamma_{ij} = 1, \sum_{i=1}^{m} \gamma_{ij} < 1, \forall i, j\Big\}. \quad (3)$$

Note that since a younger plant is mapped to an older plant in the temporal sequence, the number of skeleton nodes in the younger plant is likely to be less than the older plant. Also, some nodes might not find a mapping to the next temporal sequence. In order to find the optimal cost of matching the nodes, first we define some attributes to each node. Let $\mathcal{P}_a$ and $\mathcal{P}_b$ are the set of skeleton nodes having vertices $(\mathcal{V}_1^a, \cdots, \mathcal{V}_l^a)$ and $(\mathcal{V}_1^b, \cdots, \mathcal{V}_m^b)$. In general for any vertex $\mathcal{V}_y^x$, we assign the total degree of the node as $\alpha_y^x$, geodesic distance of the vertex from the root node (at the bottom) as $h_y^x$, and the set of other nodes that are immediate neighbour of the vertex as $d_y^x$. Basically the attribute of vertex $\mathcal{V}_y^x$ is multidimensional, which are used as features of the vertex. We also assign an index to each branch of the skeleton and store the branch index for every node. All the vertex features of the two skeletons are stored in matrices $\mathcal{C}^a$ and $\mathcal{C}^b$ respectively. Then the cosine similarity is computed to find the cost of matching the vertices between the two skeleton graphs. The similarity matrix $\mathcal{M}$ is computed as,

$$\mathcal{M} = \frac{\mathcal{C}^a \cdot \mathcal{C}^b}{||\mathcal{C}^a|| \cdot ||\mathcal{C}^b||}. \quad (4)$$

From the similarity matrix $\mathcal{M}$, nodes with maximum similarity are considered as potential matches at the initial stage. We find the set of $p$ similar pairs of nodes as $\{\mathcal{M}_s^a, \mathcal{M}_s^b\}_{s=1}^p$. Although most of the matches obtained from the cosine similarity metric are valid, however there can be some particular cases which needs to be addressed explicitly. More specifically, we eliminate bad matches (or refine the matches) by imposing the following two constraints. First, all nodes with same branch index in $\mathcal{C}^a$ should map to same branch index in $\mathcal{C}^b$. Basically in order to avoid mapping of multiple nodes to a single node, each mapped node (in the grown up plant) is eliminated from the list of unmapped nodes after each map. Second, any two nodes with different branch index in $\mathcal{C}^a$ should not map to same branch index in $\mathcal{C}^b$. Algorithm 3 shows the steps of correspondence estimation.

**Algorithm 3** Correspondence Estimation between Skeleton Graphs

**Require:** $\mathcal{P}_a = \{\mathcal{V}_1^a, \cdots, \mathcal{V}_l^a\}$, $\mathcal{P}_b = \{\mathcal{V}_1^b, \cdots, \mathcal{V}_m^b\}$ where $\mathcal{V}_y^x = \{\alpha_y^x, h_y^x, d_y^x\}$
**Ensure:** Mapping between $\tilde{\mathcal{P}}_a$ and $\tilde{\mathcal{P}}_b$ where $\tilde{\mathcal{P}}_a \subseteq \mathcal{P}_a$ and $\tilde{\mathcal{P}}_b \subseteq \mathcal{P}_b$
Initialize matrices $\{\mathcal{C}^a\}_{3\times l}$ and $\{\mathcal{C}^b\}_{3\times m}$ ▷ l, m are number of skeleton nodes in $\mathcal{P}_a$ and $\mathcal{P}_b$ respectively
Initialize matrix $\{\mathcal{M}\}_{l\times m}$ to 0
**for** $j = 0$ to $l$, $k = 0$ to $m$ **do**
  $\mathcal{C}_{1j}^a \leftarrow \mathcal{C}_{1j}^a \oplus \alpha_j^a$, $\mathcal{C}_{1k}^b \leftarrow \mathcal{C}_{1k}^b \oplus \alpha_k^b$ ▷ $\oplus$ denotes appending operation
  $\mathcal{C}_{2j}^a \leftarrow \mathcal{C}_{2j}^a \oplus h_j^a$, $\mathcal{C}_{2k}^b \leftarrow \mathcal{C}_{2k}^b \oplus h_k^b$
  $\mathcal{C}_{3j}^a \leftarrow \mathcal{C}_{3j}^a \oplus d_j^a$, $\mathcal{C}_{3k}^b \leftarrow \mathcal{C}_{3k}^b \oplus d_k^b$
**end for**
$\mathcal{M} \leftarrow$ CosineSimilarity($\mathcal{C}^a, \mathcal{C}^b$)
$\{\mathcal{M}_s^a, \mathcal{M}_s^b\}_{s=1}^p \leftarrow$ findSimilarPairs($\mathcal{M}$)
$(\tilde{\mathcal{M}}_s^a, \tilde{\mathcal{M}}_s^b) \leftarrow$ refinePairs($\mathcal{M}_s^a, \mathcal{M}_s^b$) ▷ Section III
Return the mapping $(\tilde{\mathcal{M}}_s^a, \tilde{\mathcal{M}}_s^b)$

## IV. Results

### A. Datasets

We have performed experiments on 3 datasets comprising of both real and synthetic plants. The first dataset is the Pheno4D [27] having a number of temporal sequences of Maize and Tomato plant point cloud data. The second dataset consists of several temporal sequence of *Arabidopsis thaliana* plants [11]. We have also generated challenging

Fig. 4. Skeletonization results of Maize plant (top row), synthetic plant (middle row), and *Arabidopsis* plant (bottom row). The first column shows the original plants and subsequent columns shows results from the proposed method, Xu *et al.* [18], L1 median [26], and Laplacian [23] (results of L1 median and Laplacian are obtained after tuning the parameters and finding the best value).

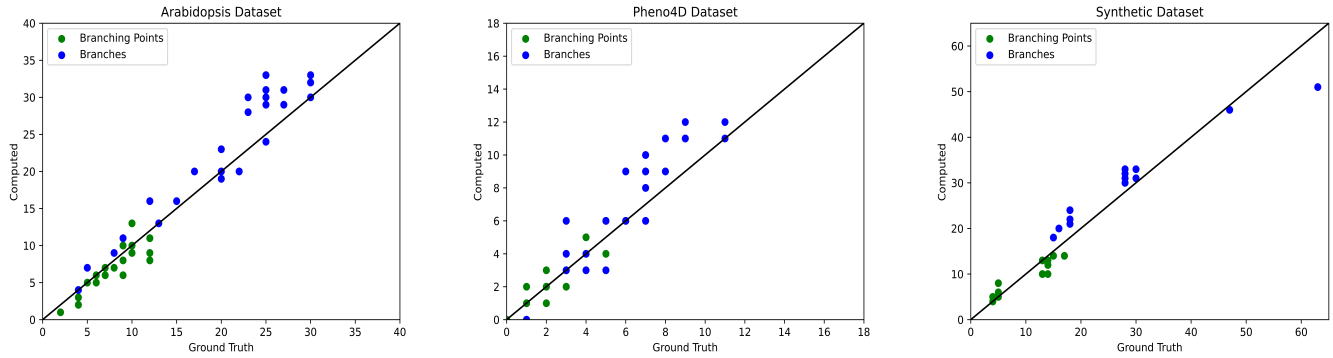


Fig. 5. Quantitative results of phenotype parameter extraction (number of branching points and number of branches) on different datasets. The $x$ axis is the ground truth, whereas $y$-axis is the computed value from the skeleton graph. Note that each data point does not necessarily mean one plant sample since there can be multiple plants having the same number of branches and the computed number of branches can also be the same.

cases of plant structures by simulating virtual plant models [29], [30] and created a dataset of 10 sequences of spatio-temporal point cloud data (dataset is available in [31]). We have also considered another challenging synthetic dataset of PypeTree [28] to show the effectiveness of the skeletonization algorithm. Some sample plants from the datasets are shown in Figure 3.

### *B. Skeletonization Results*

First, we demonstrate the results of skeletonization method in both qualitative and quantitative manner. The results are compared with 3 state-of-the-art skeletonization algorithms: Xu *et al.* [18], L1 median [26], and Laplacian [24]. We have implemented the method of Xu *et al.* [18], and the code of rest of the two methods are available publicly. Figure 4 shows qualitative results of a Maize plant and a synthetic plant with complex canopy coverage having extremely thin branches at the top. Our skeletonization method has been able to capture the geometric details of the original point cloud data in comparison with the other state-of-the-art. For all the datasets, two phenotypic parameters (number of branches and branching points) are computed automatically from the skeleton graphs. In Figure 5, the quantitative results are shown where the horizontal axis denotes the ground truth and the vertical axis denotes the computed value. As can be seen from the graphs, the computed values are close to the ground truth measurements.

Figure 6 shows the running time graph of skeletonization algorithm in comparison with [26] and [24], while [18] is not compared due to the heuristics in the original paper that could not be implemented for all datasets. The proposed method has computational time complexity of $\mathcal{O}(N^3)$ (where $N$ is the number of points).

### *C. Correspondence Estimation Results*

Next we demonstrate the results of correspondence estimation and plant organ tracking over the temporal sequence. Figure 1 and 7 show examples of correspondence over a time period of plant growth. The scans are taken several days apart from each other, thus resulting in large deformation of the point cloud. Our approach is able to find meaningful correspondences among different organs of the plant (shown in black lines).

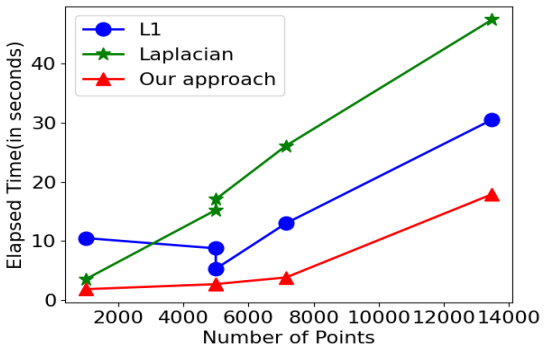


Fig. 6. Comparison of execution time of different algorithms for different number of points.

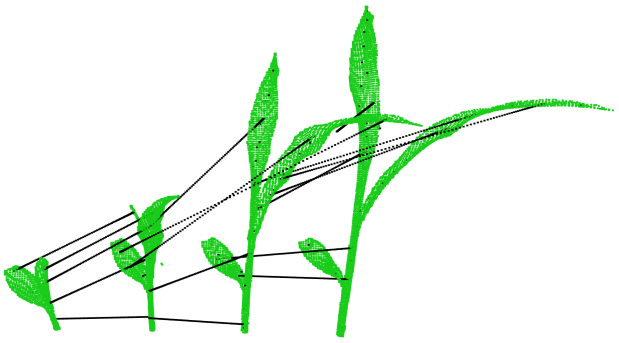

Fig. 7. An example of 4D correspondence estimation of a growing Maize plant. Black lines show the correspondence between organs.

We also perform quantitative analysis of organ tracking in the time series data. The accuracy of organ tracking is quantified by the number of correct correspondence matches at each organ level. Let us say that a plant $P_i$ has a set of organs $(O_1^i, O_2^i, \cdots, O_m^i)$ where $m$ is the number of organs in $P_i$, and a plant $P_j$ in the next temporal sequence has the set of organs $(O_1^j, O_2^j, \cdots, O_n^j)$ where $n$ is the number of organs in $P_j$. Then if a correspondence maps between $O_k^i$ and $O_l^j$ where $k = l$, then the match is considered as a true match. Similarly if $k \neq l$, then the match is considered as false match. We have computed the precision percentage (true matches / true matches + false matches) for all the three datasets. For all the correspondences, we have manually identified whether a correspondence is matching to the same organ in the next sequence. The quantitative results of precision percentage for all the datasets are shown in Table I. In general we observe that the *Arabidopsis* data is the most challenging because of close branches almost touching with each other, bent main stem, and thin structure.

TABLE I
QUANTITATIVE RESULTS OF ORGAN MATCHING IN THE TEMPORAL SEQUENCE FOR EACH DATASET

| | **Pheno4D** | **Arabidopsis** | **Synthetic** |
|---|---|---|---|
| Precision % | 86.2 % | 81.4 % | 91.3 % |

We also report tracking the growth of internodes (the distance between the branching points) over the time series data. The change of internode distances over several days (day #1, #3, #7, #10, #16, #19) are shown in Figure 8 for *Arabidopsis* data (since there are many branches on the main stem in *Arabidopsis* plant, we show the result for this plant for better visualization).

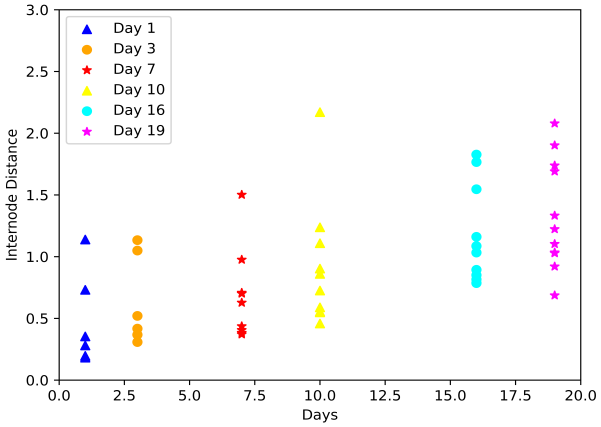


Fig. 8. Growth of internodes of growing plant over several days in *Arabidopsis* data. Each color represents the internodes of a single day. Note that as the plant grows, the number of internodes also increases.

The $x$-axis denotes the number of days and the $y$-axis denotes the internode distances. The growth of internodes can be visualized from the plot. We have also conducted experiments on tracking branch lengths, and obtained similar results like the internode growth. In general, the growth rate of *Arabidopsis* is seen to be higher than Maize plants. This also complies with the fact that *Arabidopsis* plants have shorter life cycle, and higher rate of growth of organs than Maize plants.

## V. CONCLUSION

In this paper, we have presented an approach of 4D correspondence estimation of growing plants. The approach is based on two stages. In the first stage, an efficient algorithm is developed for robust skeleton extraction and graph formation, whereas in the second stage, the skeleton graphs are matched in the time series data. Several challenging cases of thin branches and deformation were considered. Experiments on varieties of plants including real and synthetic datasets are performed. Some immediate future research direction as extension of this work can be to address the issues of large scale occlusion/missing data, considering extremely thin structures (like conifer tree), and also to address noisy point cloud data (which is often the case for many real world data acquisition). We have considered small plants having elongated organs. Considering the cases of large trees having big leaves and complex shaped organs will be challenging. Also we have not considered tracking of targeted organs (e.g. some particular fruits, flowers, etc), considering which will be interesting and beneficial to agricultural automation.

## REFERENCES


[1] C. Costa, U. Schurr, F. Loreto, P. Menesatti, and S. Carpentier, "Plant phenotyping research trends, a science mapping approach," *Frontiers in plant science*, vol. 9, p. 1933, 2019.

[2] J. Weyler, A. Milioto, T. Falck, J. Behley, and C. Stachniss, "Joint plant instance detection and leaf count estimation for in-field plant phenotyping," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3599–3606, 2021.

[3] F. Magistri, N. Chebrolu, J. Behley, and C. Stachniss, "Towards in-field phenotyping exploiting differentiable rendering with self-consistency loss," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 13 960–13 966.

[4] J. Weyler, F. Magistri, P. Seitz, J. Behley, and C. Stachniss, "In-field phenotyping based on crop leaf and plant instance segmentation," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2022, pp. 2725–2734.

[5] H. Freeman, E. Schneider, C. H. Kim, M. Lee, and G. Kantor, "3d reconstruction-based seed counting of sorghum panicles for agricultural inspection," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9594–9600.

[6] A. Silwal, T. Parhar, F. Yandun, H. Baweja, and G. Kantor, "A robust illumination-invariant camera system for agricultural applications," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3292–3298.

[7] A. Chaudhury, C. Ward, A. Talasaz, A. G. Ivanov, M. Brophy, B. Grodzinski, N. P. A. Hüner, R. V. Patel, and J. L. Barron, "Machine vision system for 3d plant phenotyping," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 6, pp. 2009–2022, 2018.

[8] I. Ziamtsov and S. Navlakha, "Machine learning approaches to improve three basic plant phenotyping tasks using three-dimensional point clouds," *Plant physiology*, vol. 181, no. 4, pp. 1425–1440, 2019.

[9] A. Chaudhury, P. Hanappe, R. Azaïs, C. Godin, and D. Colliaux, "Transferring pointnet++ segmentation from virtual to real plants," in *Computer Vision in Plant Phenotyping and Agriculture (CVPPA), in Conjunction with International Conference on Computer Vision (ICCV)*, 2021.

[10] F. Magistri, N. Chebrolu, and C. Stachniss, "Segmentation-based 4d registration of plants point clouds for phenotyping," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2433–2439.

[11] H. Pan, F. H. Wheeler, J. Charlaix, and D. Colliaux, "Multi-scale space-time registration of growing plants," in *2021 International Conference on 3D Vision (3DV)*, 2021, pp. 310–319.

[12] K. Heiwolt, C. Öztireli, and G. Cielniak, "Statistical shape representations for temporal registration of plant components in 3d," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9587–9593.

[13] A. Riccardi, S. Kelly, E. Marks, F. Magistri, T. Guadagnino, J. Behley, M. Bennewitz, and C. Stachniss, "Fruit tracking over time using high-precision point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 9630–9636.

[14] N. Chebrolu, T. Läbe, and C. Stachniss, "Spatio-temporal non-rigid registration of 3d point clouds of plants," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 3112–3118.

[15] A. Verroust and F. Lazarus, "Extracting skeletal curves from 3d scattered data," in *Proceedings Shape Modeling International'99. International Conference on Shape Modeling and Applications*, 1999, pp. 194–201.

[16] T. K. Dey and J. Sun, "Defining and computing curve-skeletons with medial geodesic function," in *Symposium on geometry processing*, vol. 6, 2006, pp. 143–152.

[17] A. Sharf, T. Lewiner, A. Shamir, and L. Kobbelt, "On-the-fly curve-skeleton computation for 3d shapes," in *Computer Graphics Forum*, vol. 26, no. 3, 2007, pp. 323–328.

[18] H. Xu, N. Gossett, and B. Chen, "Knowledge and heuristic-based modeling of laser-scanned trees," *ACM Transactions on Graphics (TOG)*, vol. 26, no. 4, 2007.

[19] A. Bucksch, R. Lindenbergh, and M. Menenti, "Skeltre: Robust skeleton extraction from imperfect point clouds," *The Visual Computer*, vol. 26, no. 10, pp. 1283–1300, 2010.

[20] Y. Livny, F. Yan, M. Olson, B. Chen, H. Zhang, and J. El-Sana, "Automatic reconstruction of tree skeletal structures from point clouds," in *ACM SIGGRAPH Asia*, 2010, pp. 1–8.

[21] A. Runions, B. Lane, and P. Prusinkiewicz, "Modeling trees with a space colonization algorithm," *Eurographics Workshop on Natural Phenomena*, vol. 7, no. 63-70, p. 6, 2007.

[22] O. K. Au, C. Tai, H. Chu, D. Cohen-Or, and T. Lee, "Skeleton extraction by mesh contraction," *ACM transactions on graphics (TOG)*, vol. 27, no. 3, pp. 1–10, 2008.

[23] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," in *ACM SIGGRAPH*, 2009, pp. 1–9.

[24] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via laplacian based contraction," in *Shape Modeling International Conference*. IEEE, 2010, pp. 187–197.

[25] S. Wu, W. Wen, B. Xiao, X. Guo, J. Du, C. Wang, and Y. Wang, "An accurate skeleton extraction approach from 3d point clouds of maize plants," *Frontiers in plant science*, vol. 10, p. 248, 2019.

[26] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud." *ACM Transactions on Graphics*, vol. 32, no. 4, pp. 65–1, 2013.

[27] D. Schunck, F. Magistri, R. A. Rosu, A. Cornelißen, N. Chebrolu, S. Paulus, J. Léon, S. Behnke, C. Stachniss, H. 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*, vol. 16, no. 8, 2021.

[28] S. Delagrange, C. Jauvin, and P. Rochon, "Pypetree: A tool for reconstructing tree perennial tissues from point clouds," *Sensors*, vol. 14, no. 3, pp. 4271–4289, 2014.

[29] F. Boudon, C. Pradal, T. Cokelaer, P. Prusinkiewicz, and C. Godin, "L-py: an l-system simulation framework for modeling plant architecture development based on a dynamic language," *Frontiers in plant science*, vol. 3, p. 76, 2012.

[30] A. Chaudhury, F. Boudon, and C. Godin, "3d plant phenotyping: All you need is labelled point cloud data," in *ECCV Workshop on Computer Vision Problems in Plant Phenotyping*, 2020, pp. 244–260.

[31] https://www.kaggle.com/datasets/sharmisthabp/synthetic-plant-dataset-for-computer-vision.