frontiers in Plant Science

ORIGINAL RESEARCH
published: 16 June 2020
doi: 10.3389/fpls.2020.00773

# Skeletonization of Plant Point Cloud Data Using Stochastic Optimization Framework

*Ayan Chaudhury*[1,2] *and Christophe Godin*[1,2]*

[1] *INRIA Grenoble Rhône-Alpes, Team MOSAIC, Lyon, France,* [2] *Laboratoire Reproduction et Développement des Plantes, Univ Lyon, ENS de Lyon, UCB Lyon 1, CNRS, INRA, Lyon, France*

OPEN ACCESS

**Edited by:**
*Matti Mõttus,*
*VTT Technical Research Centre of Finland Ltd, Finland*

**Reviewed by:**
*Eetu Puttonen,*
*National Land Survey of Finland, Finland*
*Pasi Raumonen,*
*University of Tampere, Finland*
*Weiliang Wen,*
*Beijing Research Center for Information Technology in Agriculture, China*

***Correspondence:**
*Christophe Godin*
*christophe.godin@inria.fr*

**Specialty section:**
*This article was submitted to Technical Advances in Plant Science, a section of the journal Frontiers in Plant Science*

**Received:** *15 February 2020*
**Accepted:** *15 May 2020*
**Published:** *16 June 2020*

**Citation:**
*Chaudhury A and Godin C (2020) Skeletonization of Plant Point Cloud Data Using Stochastic Optimization Framework. Front. Plant Sci. 11:773. doi: 10.3389/fpls.2020.00773*

Skeleton extraction from 3D plant point cloud data is an essential prior for myriads of phenotyping studies. Although skeleton extraction from 3D shapes have been studied extensively in the computer vision and graphics literature, handling the case of plants is still an open problem. Drawbacks of the existing approaches include the zigzag structure of the skeleton, nonuniform density of skeleton points, lack of points in the areas having complex geometry structure, and most importantly the lack of biological relevance. With the aim to improve existing skeleton structures of state-of-the-art, we propose a stochastic framework which is supported by the biological structure of the original plant (we consider plants without any leaves). Initially we estimate the branching structure of the plant by the notion of $\beta$-splines to form a *curve tree* defined as a finite set of curves joined in a tree topology with certain level of smoothness. In the next phase, we force the discrete points in the curve tree to move toward the original point cloud by treating each point in the curve tree as a center of Gaussian, and points in the input cloud data as observations from the Gaussians. The task is to find the correct locations of the Gaussian centroids by maximizing a likelihood. The optimization technique is iterative and is based on the Expectation Maximization (EM) algorithm. The E-step estimates which Gaussian the observed point cloud was sampled from, and the M-step maximizes the negative log-likelihood that the observed points were sampled from the Gaussian Mixture Model (GMM) with respect to the model parameters. We experiment with several real world and synthetic datasets and demonstrate the robustness of the approach over the state-of-the-art.

**Keywords: skeletonization, point cloud, curve tree, spline, stochastic optimization, Gaussian mixture model, expectation maximization**

## 1. INTRODUCTION

Automated analysis of phenotyping traits of plants using imaging techniques are becoming off-the-shelf tool for botanical analyses these days (Spalding and Miller, 2013; Tardieu et al., 2017). Although high throughput 2D imaging based phenotyping systems have shown promising results on the accuracy and precision for many cases (Brichet et al., 2017; Zhang et al., 2017; Choudhury et al., 2019), but these systems suffer from the inherent limitations of 2D image based analysis techniques. In recent years, 3D point cloud based analysis is getting extremely popular in phenotyping and agricultural applications (Vázquez-Arellano et al., 2016). Typical applications of point cloud based phenotyping include plant organ segmentation (Ziamtsov and Navlakha, 2019), robotic branch pruning (Chattopadhyay et al., 2016), automated growth analysis (Chaudhury et al., 2019), etc. Many of these applications require skeleton structure of the input point cloud data as a prior for further processing. Also, as a skeleton is a compact representation of the original object, dealing with skeleton requires less computational overhead than dealing with the original point cloud data. That is why skeletons are widely used for varieties of shape analysis tasks in different application areas (Cornea et al., 2007).

In general, a skeleton is a thin structure obtained from an object, which encodes the topology and basic geometry of the object. Ideally, the skeleton should follow the exact centerline of the object. That means within a local neighborhood, the distance from each skeleton point to the enclosing shape boundaries should be the same. Although skeleton extraction from 2D images is a well-studied area, 3D point cloud skeletonization is still an open problem (Tagliasacchi et al., 2016). For the case of plants, skeleton extraction is extremely challenging due to their complex geometry, thin structure, missing data due to self-occlusion, etc. State-of-the-art algorithms (Tagliasacchi et al., 2009; Cao et al., 2010; Huang et al., 2013) which are proposed as general skeletonization techniques for regular 3D objects, often produce degraded results when applied to complex plant branching structures. Skeletonization of 3D plant point cloud data is a little worked problem, and only a handful of techniques have been proposed in recent years. These techniques can be broadly classified into two categories: (i) initial skeleton construction, and (ii) skeleton refinement.

The first type of algorithm refers to skeleton construction from the input point cloud data. A notable work in this category is the method proposed by Xu et al. (2007). Initially, a *Riemannian graph* is constructed from the point cloud by connecting the points using nearest neighbor strategy. Then the whole point cloud is clustered based on graph adjacency information and quantized shortest path lengths from the root to all points in the cloud. The center of each cluster is assigned as a skeleton point and the edges are defined as the connection of cluster centers based on their spatial locations. One problem of the technique is that, the skeleton does not maintain the centeredness criteria, and results in a zigzag structure near the junction point where two or more branches meet. Also, the computed skeleton points sometimes end up being outside the boundary of the point cloud (we discuss about these issues in the next section). Bucksch et al. (2009) proposed a solution of these type of problems by subdividing the point cloud into octree cells and used some local heuristics to form a skeleton graph from the cells. Zhang et al. (2015) later applied this technique locally to individual branches. However, these methods rely on several heuristic assumptions and require many parameters to be tuned to obtain the desired result. On a different type of approach, *particle flow* based techniques (Rodkaew et al., 2003; Zeng et al., 2007) are built on the motivation on the process of transport and exchange of energy and water between root, branches and leaves of a plant. However, the skeleton is not guaranteed to follow the actual geometry of the input point cloud data, and thus suffers from biological irrelevance of the results. This technique is mainly used in computer graphics applications. *Space colonization* algorithm (Runions et al., 2007; Preuksakarn et al., 2010) extracts skeleton from input data by an iterative technique to *eat-up* the points in the cloud starting from the root node at the bottom. The overall technique is a local optimization based strategy. Although the technique can produce visually pleasing skeleton structure, the algorithm might result in creating branching structure in the wrong directions, and there is no mechanism to perform backtracking to correct the biological irrelevance of the branches.

The second type of approach of skeletonization is based on the motivation to improve the initial skeleton. Livny et al. (2010) proposed a series of optimization techniques to smooth the branches of a skeleton in a realistic manner. However, the optimization strategy does not involve prior botanical knowledge of plants. Branch smoothing is performed independently without taking into consideration the original data, and can produce over-smoothing or under-smoothing of branches which suffer from botanical inconsistency of the results. In a similar line of work, Zhen et al. (2016) proposed a strategy to refine an existing skeleton to handle the case of occlusion and missing data. The approach is based on a combined local and global optimization strategy, where the coarse skeleton is pushed to move toward the original point cloud, and the original point cloud is forced to contract toward the skeleton points. The method is explicitly built to handle the occlusion cases and overlooks factors like zigzag problem and biological irrelevance of the skeleton. In a recent work, Wu et al. (2019) proposed a skeleton refinement technique for Maize plants. The issue of centeredness criteria and zigzag problem is explicitly addressed. The refinement technique is based on a local neighborhood based approximation strategy and some heuristics are used. The method is tested only on a single plant species. Other types of recent skeletonization techniques are shown to be successful for some applications (Jin et al., 2016; Tabb and Medeiros, 2018), where it is assumed that the point cloud is voxelized along with voxel connectivity information, instead of considering raw point cloud data without any connectivity information. Geometric model fitting methods (Liang et al., 2013; Raumonen et al., 2013; Hackenberg et al., 2014) fit cylinder models to the tree branches. Although the methods do not explicitly consider problem of skeleton extraction from point cloud data, but ideally computing the central axes of the fitted cylinders can be performed to extract the skeleton. However, no experimental results are reported in this aspect.

We propose a skeletonization technique that belongs to the second category. The aim is to improve an existing skeleton obtained from the state-of-the-art by maintaining the biological relevance with the input point cloud data. Initially we represent a coarse skeleton as a tree structure which consists of superposition of parametric curves having uniform density of discrete points. Then we propose a stochastic optimization technique to transform the skeleton points so that the transformed points maintain the centeredness criteria as well as the biological relevance with the input data. Existing skeleton refinement techniques typically aim at improving an initial coarse skeleton by optimizing an objective function. Often the objective function is constructed using biological prior, and the optimization does not make use of the original input point cloud data (Livny et al., 2010). Other approaches (Zhen et al., 2016) deform the original

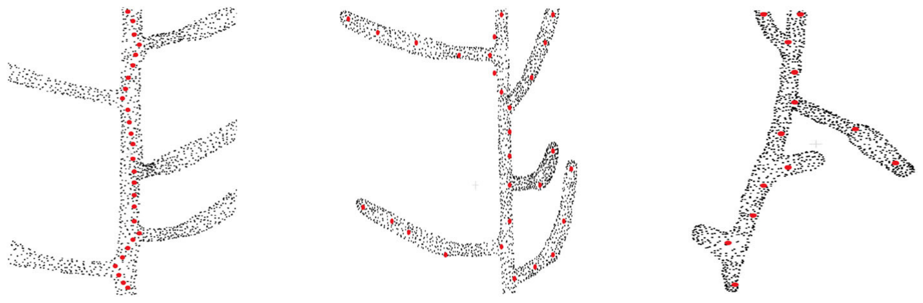

**FIGURE 1 |** Problems with the existing approaches on skeletonizing plant point cloud data. Original point cloud is shown as black point cloud and the skeleton points are shown as red points. **(Left)** The problem of zigzag structure, where the skeleton does not follow the centerline of the stem and tends to deviate toward the branching point (Xu et al., 2007) (only the main stem skeleton is shown in the figure). **(Middle)** The problem of biologically irrelevant skeleton points which falls beyond the boundary of the input data (shown at the top part), and inability to capture the geometric details for some branches (Delagrange et al., 2014; Ziamtsov and Navlakha, 2019). **(Right)** Overlooking tiny geometrical structures (Xu et al., 2007).

point cloud data during the optimization process. However, these types of approaches do not guarantee the optimized skeleton to be geometrically consistent with the original point cloud data. One of the main strength of our approach is the exploitation of the input point cloud data during the optimization process. This explicitly helps the skeleton to retain biologically supported structure.

## 2. PROBLEM STATEMENT

The motivation of this work is to extract the skeleton from plant point cloud data. Before we discuss our model in the next section, we first demonstrate the limitations of the existing approaches on real datasets. We identify the following 3 problems associated with the existing techniques, as shown visually in **Figure 1**.

### 2.1. The *Zigzag* Structure Problem

One of the typical problems associated with skeletonizing plant point cloud data is the deviation of skeleton points from the centerline of the branch toward the junction point where two or more branches are joined Xu et al. (2007), thus creating a zigzag type skeleton as shown in the left of **Figure 1**. The zigzag problem occurs mainly because of the local nature of the algorithm. During the clustering phase, the data is quantized into several levels, and the center of each cluster is computed as a skeleton point. For a simple branch, the cluster center is computed at the center of the branch. However, for the case of a bifurcation, the cluster center tends to shift toward the bifurcating branch, thus resulting in a zigzag like structure. Ideally the skeleton points should follow the exact centerline of the branch representing the actual geometry of the plant. However, the zigzag skeleton structure is a wrong approximation of the plant geometry, which results in wrong phenotyping parameter estimation in quantitative measurement of phenotypes such as internode distance, branch length, etc. Although this problem is handled locally in Wu et al. (2019), the method is based on many heuristics to skeletonize a handful of Maize plant point cloud data.

### 2.2. Invalid and Inaccurate Geometry Estimation Problem

As shown in the middle of **Figure 1**, sometimes skeleton points do not lie within the enclosing boundary of the original point cloud data (Delagrange et al., 2014; Ziamtsov and Navlakha, 2019). This results in invalid geometry estimation of the input data. For example, in the upper part of the figure we can see the red dots which are located outside the boundary of the point cloud. Also for the case of curved branches, insufficient number of skeleton points are not able to represent the actual geometry of the branch. This is shown at the bottom of the figure where the curved branches are not represented by the skeleton points. One way to handle this problem might be to generate more skeleton points in the curved areas using interpolation or similar point approximation strategies. However, the generated points will be independent of the original point cloud data, and will fail to reconstruct the original geometry of the branch.

### 2.3. Inability to Handle Tiny Structures

Tiny branches are often treated as noise in the point cloud data, and the extracted skeleton fails to represent these branches. This is shown at the right of **Figure 1**. Two tiny branches are completely overlooked in terms of skeleton representation of the input data (Xu et al., 2007).

## 3. BUILDING AND OPTIMIZING THE SKELETON

### 3.1. Skeleton Intialization

Initially we obtain a coarse skeleton graph of the input point cloud $\mathbb{P} = \{\mathbf{p}_1, \cdots, \mathbf{p}_n\} \in \mathbb{R}^3$ by the method of Xu et al. (2007). Note that other skeletonization algorithms can also be used to obtain the coarse skeleton, but we have used this method because the algorithm is simple, fast, works reasonably well, and the open source implementation is available[1]. Next we remove the cycles in the graph by the method of Yan et al. (2009). The resultant skeleton point set is a rooted tree graph embedded in $\mathbb{R}^3$. In this

[1]https://github.com/fredboudon/plantscan3d

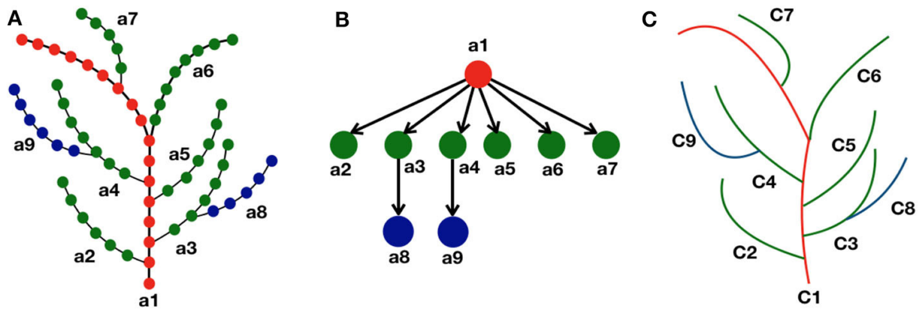

**FIGURE 2 |** Representation of skeleton tree graph as a curve tree via quotient graph. **(A)** Each branch ($a1, \cdots, a9$) in the skeleton tree represents an axis where the discrete points are shown as colored dots using a hierarchical color convention. The main stem is represented as red dots, next level branches as green dots, and further level branches as blue dots. The successor of a node of certain color is colored with the same color. **(B)** Each node in the quotient graph represents each branch (or axis) of the skeleton graph, and the directed edges represent their hierarchical relationship in the skeleton structure. **(C)** Corresponding curve tree where each branch in the skeleton graph is approximated as a parametric curve ($C1, \cdots, C9$).

work, we represent the input skeleton and its branching structure as an *axial tree* (Prusinkiewicz and Lindenmayer, 1990; Godin and Caraglio, 1998).

Formally, let $\mathcal{G} = (V, E)$ be a graph where $V = \{v_i\}_{1\leq i\leq I}$ (where I is an integer $\geq 1$) is the set of vertices, and $E = (v_i, v_j)$ is the set of edges connecting to the ordered pairs of vertices $v_i$ and $v_j$. The vertex $v_j$ is the child of $v_i$ denoted as, $\text{child}(v_i) = \{v_j \in V | (v_i, v_j) \in E\}$, and the vertex $v_i$ is the parent of $v_j$ denoted as $v_i = \text{parent}(v_j)$. A rooted tree graph is a graph $\mathcal{G}$ that contains no cycle, only one connected component, and there exists a unique vertex in $V$, called the *root* that has no parent. The *descendants* of a vertex $v_i \in V$ is defined as the set of vertices that belong to the branching system starting from $v_i$ excluding itself.

**Definition 1:** Axial Tree, adapted from Godin and Caraglio (1998): An *Axial Tree* $\mathcal{T}$ is defined as a tree graph along with a *successor* function *succ* associated with each vertex in the graph defined as,

$$\begin{aligned} \text{succ} : V &\to V, \\ v_i &\mapsto v_{i'}, \text{where } v_{i'} \in \text{child}(v_i), \end{aligned}$$

where $v_{i'}$ is called the successor of $v_i$.

From the definition, we see that all the vertices $v_i \in V$ in an axial tree, there exists at most one successor $v_{i'} \in V$. Note that there may be vertices that do not have a successor in $V$ (such as leaf nodes in particular). Maximal set of vertices connected by successor relationship are paths in $\mathcal{T}$ (called the *axes* of $\mathcal{T}$). For example in **Figure 2A**, the axes are denoted as $a1, \cdots, a9$. In this paper, a skeleton is thus represented by an axial tree $\mathcal{T}$ for which the vertices correspond to the points in the skeleton, and the edges as the ordering of these points within their axis. The 3D coordinates $\mathbf{s}_i$ of each point is attached to the corresponding vertex $v_i$.

In a given axis $a = \{v_1, \cdots, v_k\}$ (with slight abuse of notation), we define the *bearing vertex* $w$ as, $w = b(a) \Leftrightarrow w = \text{parent}(v_1)$, if it exists. The *order* relationship between the vertices derived from their order on the path in the axis is defined as, $v_i < v_j \Leftrightarrow v_i$ is an antecedent of $v_j$.

## 3.2. Parametric Representation of the Tree Skeleton

Now we want to define for each axis in the axial tree a parametric model representing its geometry. For this, we first define a conceptual model representing the branching geometry of the skeleton, and then explain how it is constructed from the data. Let us first introduce the following definitions.

In an axial tree $\mathcal{T}$, we say that two vertices $v_i$ and $v_j$ are in *equivalence relationship* (denoted as $v_i \equiv v_j$) if and only if $v_i$ and $v_j$ belong to the same axis of an axial tree. This allows us to define the notion of quotient tree graph as follows.

**Definition 2:** Quotient Tree Graph, adapted from Godin and Caraglio (1998): Given an axial tree graph $\mathcal{T} = (V, E)$, a *Quotient Tree Graph* $\overline{\mathcal{T}} = (\overline{V}, \overline{E})$ is the quotient graph of the axial tree graph $\mathcal{T}$, which is made of the axes of $\mathcal{T}$, and is also a tree graph.

The vertices in an axial tree having equivalence relationship with each other, are collapsed to a single node in the quotient graph (**Figure 2B**).

Now consider a family of curves $\mathscr{C}$ defined in $\mathbb{R}^3$. With each vertex $v_a \in \overline{V}$ in the quotient graph $\overline{\mathcal{T}}$, we associate a curve $\mathcal{C}_a \in \mathscr{C}$ defined as,

$$\begin{aligned} \mathcal{C}_a : [0,1] &\to \mathbb{R}^3, \\ u &\mapsto \mathcal{C}_a(u), \end{aligned}$$

where $u \in [0, 1]$ is the parameter of the curve. Curves on the axes must respect the following attachment conditions to one another. For any two vertices $a_i, a_j \in \overline{V}$ such that $a_j = \text{parent}(a_i)$, then the following condition holds:

$$\exists u'_0 \in [0,1], \text{ s.t. } \mathcal{C}_{a_i}(0) = \mathcal{C}_{a_j}(u'_0). \tag{1}$$

Ideally, we want to represent the skeleton as a union of curves, one curve is being attached with a vertex of the quotient tree graph, where the order of the branching points in the skeleton is the order of vertices in the quotient graph for the attachment of the curves. In order to achieve this, the following condition must hold true in $\overline{\mathcal{T}}$. Consider the axes $a_i, a_j, a_k$, where $a_i = \text{parent}(a_j)$ and $a_i = \text{parent}(a_k)$. Let $w_j$ and $w_k$ be the respective bearing vertices for the axes $a_j$ and $a_k$. Then the following condition holds true:

$$\left.\begin{array}{l} w_j \leq w_k \\ \mathcal{C}_{a_j}(0) = \mathcal{C}_{a_i}(u'_0) \\ \mathcal{C}_{a_k}(0) = \mathcal{C}_{a_i}(u''_0) \end{array}\right\} \Rightarrow u'_0 \leq u''_0. \tag{2}$$

**Definition 3:** Curve Tree: A *curve tree* $\mathcal{T}_C = (\mathcal{T}, g)$ is an axial tree $\mathcal{T}$ augmented with a mapping $g$ that attaches one parametric curve to each vertex of $\overline{\mathcal{T}}$, and verifying the attachment conditions in Equations (1) and (2) on its quotient tree graph. The mapping $g$ is defined as,

$$g : \overline{V} \to \mathscr{C}$$
$$v_a \mapsto \mathcal{C}_a,$$

where the curve $\mathcal{C}_a$ is associated with each vertex $v_a \in \mathcal{T}$ (**Figure 2C**). In order to instantiate the curve associated with each axis $a$, we choose a family of curves that may be parameterized using the 3D points $\{v_1, \cdots, v_k\}$ of the axis $a$.

## 3.3. Estimation of the Curve Tree Parameters

From a given axial tree, we can define a curve tree by attaching a parametric curve to each axis $a$ of the quotient tree graph from a chosen family of parametric curves $\mathscr{C}$.

In this work, we choose to model curves representing axes using the family of splines. Splines are powerful mathematical tools to approximate curves and surfaces. In the general case, a spline curve $\mathcal{S}(u)$ defined by $(n+1)$ control points[2] $P_0, P_1, \cdots, P_n$ can be defined as,

$$\mathcal{S}(u) = \sum_{i=\lfloor nu-2 \rfloor}^{\lceil nu+2 \rceil} \mathcal{B}(nu - i) P_i, \tag{3}$$

where $\mathcal{B}(\cdot)$ is the blending function of the spline, and $u \in [0, 1]$ (Burger and Gillies, 1989). For a cubic B-spline, at most four nearest control points are used to compute the blending function for one spline point. We aim at resampling each branch of the curve tree, where the number of points in a branch can be modeled to the desired number. We have used the following strategy. Let the Euclidean distance (we are not taking into account the branch bending in the distance computation) between the endpoints of a branch is $d$ (in mm). Then we approximate the number of spline points as $\lceil d/d_0 \rceil$, where we use $d_0 = 1.0$. Depending on the application, higher or lower density of points can be achieved easily by changing the value of $d_0$ to a lower or higher value, respectively. Number of spline points can be greater than, equal to or less than the number of control points. Any curve thus can be represented as the polyline joining continuous set of spline points. Although cubic B-splines are widely used in modeling curves, we model the branch curves as $\beta$-splines, which provide an intermediate representation between approximation and interpolation and thus captures the local geometry better than cubic B-spline. Given the fact that $\beta$ splines are 1st and 2nd order continuous (Goodman, 1985), we can approximate the branches as smooth enough. The blending function of a $\beta$-spline denoted as $\beta(v)$, parameterized by $v$ is given by,

$$\beta(v) = \begin{cases} \frac{2}{\delta}(2+v)^3, & -2 \leq v \leq -1 \\ \frac{1}{\delta}((t+4s+4s^2) - 6(1-s^2)v - 3v^2(2+t+2s) \\ -2v^3(1+t+s+s^2)), & -1 \leq v \leq 0 \\ \frac{1}{\delta}((t+4s+4s^2) - 6v(s-s^3) - 3v^2(t+2s^2+ \\ 2s^3) + 2v^3(t+s+s^2+s^3)), & 0 \leq v \leq 1 \\ \frac{2}{\delta}s^3(2-v)^3, & 1 \leq v \leq 2, \end{cases} \tag{4}$$

where $v = nu - i$, $t$ is the tension parameter and $s$ is the skew parameter (Goodman, 1985; Goodman and Unsworth, 1986), which are kept constant. We used $t = 10$ and $s = 1$ for all of our experiments. While the skew parameter is always set to 1 to ensure the smoothness of the curve, we have tested different values of $t$ on several datasets. From empirical observation, we notice that keeping $t \in [5, 15]$ produces best approximation of the curve (very low error value of mean square distance between the original points and generated points), while other values of $t$ outside this range yields high error value (the error corresponding to $t = 10$ was the minimum). $\delta$ is given by $\delta = t + 2s^3 + 4s^2 + 4s + 2$. Basically the blending function is defined for 4 discrete intervals (2 in each side) around each control point. While computing the spline points near the endpoints, we use the trick of truncating the generated spline points to the endpoints of the curve in order to prevent the generated points to go beyond the boundary of the curve.

At this stage, the curve tree represents a coarse skeleton of the plant having uniform distribution of discrete points. However, the skeleton still has the zigzag structure and non-centeredness problem. We introduce a stochastic optimization approach to improve the current skeleton to follow the geometry of the original point cloud in order to obtain a biologically relevant skeleton of the input point cloud data.

## 3.4. Stochastic Modeling

Ideally, the skeleton points should follow the exact centerline of the input point cloud data. For the case where the branches of the plant are cylindrical, the skeleton should follow the central axis of the cylinder in each branch. In case of cross-sectional branches, the skeleton should go through the middle of the points by maintaining equidistance criteria from the enclosing boundaries. The initial skeleton suffers from the problem of not maintaining the centerline criteria as defined above, thus resulting in having points which are deviated from the centerline. Our goal is to improve the deviated skeleton by moving the points toward the centerline and maintain the geometry of the branching structure of the plant using a stochastic approach.

[2] By *control points*, we mean the given set of discrete input points, whereas by *spline points* we mean the generated points on the spline curve.

We frame the problem of skeleton refinement as a transformation estimation problem. We aim at moving (or transforming) the discrete points of the skeleton toward the original input point cloud data, so that the skeleton points get aligned with the centerline of the original point cloud data. We consider points in $\mathbb{P}$ (defined in section 3) as the observations of a Gaussian Mixture Model (GMM), and the centroids of the Gaussians are considered as the skeleton point set. Now in order to estimate the correct location of the centroids, we aim at finding which Gaussians the points in the skeleton point set are sampled from. Initially, the discrete points of the resampled curve tree represents the initial location of the Gaussian centroids. Then we iteratively use the Expectation Maximization (EM) algorithm to automatically approximate the optimal centroid locations and reparameterize the Gaussians in each iteration by maximizing a likelihood function. However, the problem is highly non-rigid in nature. Global transformation of points is not sufficient to achieve the optimal solution, and we aim at estimating transformation for every point in the skeleton. Exploiting the recent advancements of Gaussian Mixture Model based approaches (Myronenko and Song, 2010; Jian and Vemuri, 2011; Ma et al., 2016), we formulate the problem in a probabilistic framework.

We denote the original point cloud $\mathbb{P}$ as the *fixed* point set, and the skeleton point cloud $\mathbb{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_m\} \in \mathbb{R}^d$ of the resampled curve tree as the *moving* point set, where $d$ is the dimension of the point cloud (which is 3 in our case). In a typical case of point set registration, $m \approx n$, and there is a correspondence between most of the point pairs $(\mathbf{s}_i, \mathbf{p}_j)$. However in our case $m \ll n$, and instead of one-to-one correspondence, a group of points in the original point set corresponds to a single point in the skeleton point set. Let $\Theta$ be the set of unknown model parameters (we define $\Theta$ later). We estimate the centroid locations $\mathbb{S}^*$ from the optimal set of model parameters $\Theta^*$, which is obtained by minimizing the following negative log-likelihood,

$$\Theta^* = \underset{\Theta}{\arg\min}\ -ln(p(\mathbb{S}|\mathbb{P}, \Theta)). \tag{5}$$

This type of problem can be solved by the classical framework, where the E-step estimates which Gaussian the observed point cloud was sampled from, and the M-step maximizes the negative log-likelihood that the observed points were sampled from the GMM with respect to the model parameters. Each $\mathbf{s}_i$ is assumed to be the centroid of a Gaussian, and the corresponding $\mathbf{p}_j$'s are assumed to be normally distributed around $\mathbf{s}_i$. The probability density function is then given by,

$$p(\mathbf{p}_j|\mathbf{s}_i) = \frac{1}{(2\pi\sigma^2)^{d/2}} \exp\left(-\frac{1}{2}\left\|\frac{\mathbf{p}_j - \mathbf{s}_i}{\sigma}\right\|^2\right), \tag{6}$$

where $\sigma$ is the covariance of the Gaussian, whose initial value $\sigma_{(0)}$ is set as,

$$\sigma^2_{(0)} = \frac{1}{dmn}\sum_{i=1}^{m}\sum_{j=1}^{n}\left\|\mathbf{p}_j - \mathbf{s}_i\right\|^2. \tag{7}$$

Now we introduce outliers in the model, assuming the outlier probability to have uniform distribution for all data points. Let $\epsilon \in [0, 1]$ be the weight of the outlier distribution in the original point set (that means, each point has the probability $\epsilon/n$), and $(1-\epsilon)$ be the weight in the skeleton point set. Then the mixture model takes the form,

$$p(\mathbf{p}_j|\Theta) = \frac{\epsilon}{n} + (1-\epsilon)\sum_{i=1}^{m} p(\mathbf{p}_j|\mathbf{s}_i), \tag{8}$$

where $\Theta = \{\sigma^2, \epsilon\}$ is the set of model parameters, which we wish to estimate in EM framework in order to minimize the negative log-likelihood energy of Equation (5). Next, we propose a membership probability $p(\mathbb{S})$ for the GMM components. Instead of assigning equal probability for all the points, a weight factor is used to indicate the probability of point correspondence between the two point sets. The probability that the point $\mathbf{p}_j$ is an observation of the Gaussian of point $\mathbf{s}_i$ is defined by $\alpha_{ij}$ as the membership probability. We use a local Principal Component Analysis (PCA) based technique to estimate the membership probability as follows. Considering a local neighborhood $\mathcal{N}_i$ around the $i$-th point defined as the points within a ball of radious $r$, the $3 \times 3$ covariance matrix $\mathcal{C}_i$ is computed as,

$$\mathcal{C}_i = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{v}_k - \bar{\mathbf{v}}_k)(\mathbf{v}_k - \bar{\mathbf{v}}_k)^T, \tag{9}$$

where $v_k$ are the points in the local neighborhood of $i$-th point, N is the number of points in the neighborhood, and $\bar{v}_i = \frac{1}{N}\sum_{k\in N} v_k$. If the eigenvalues of the matrix $\mathcal{C}_i$ are $\lambda_0, \lambda_1, \lambda_2$ (where $\lambda_0 \leq \lambda_1 \leq \lambda_2$), then we compute the normalized eigenvalue of the neighborhood as,

$$\mu_i = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2}. \tag{10}$$

Note that instead of normalizing the lowest eigenvalue, other eigenvalues can also be normalized to compute $\mu_i$. We use $\mu_i$ as an estimate of the local structure of the neighborhood of $i$-th point. Similarly we compute $\mu_j$ for the other point set. Large difference of $\mu = |\mu_i - \mu_j|$ between the local neighborhood is penalized by assigning low probability, whereas small difference contributes to high probability. The membership probability that $i$-th point corresponds to $j$-th point is computed as an exponentially decaying function as,

$$\alpha_{ij} = \exp\left(-\alpha|\mu_i - \mu_j|\right), \tag{11}$$

where $\alpha$ is a tuning parameter (set to 1 for all of our experiments).

Along with the membership probability in the negative log-likelihood, we minimize the following energy by EM algorithm,

$$\mathcal{L} = -\sum_{j=1}^{n} log \sum_{i=1}^{m+1} p(\mathbf{s}_i)p(\mathbf{p}_j|\mathbf{s}_i). \tag{12}$$

*E-step*: In the E-step, the current parameter values (the *"old"* set of parameters) are used to estimate the posterior distribution $p^{\text{old}}(\mathbf{p}_j|\mathbf{q}_i)$ using Bayes' theorem as,

$$p^{\text{old}}(\mathbf{s}_i|\mathbf{p}_j) = \frac{p(\mathbf{s}_i)\,p(\mathbf{p}_j|\mathbf{s}_i)}{p(\mathbf{p}_j)}. \tag{13}$$

In order to evaluate the probability $p^{\text{old}}$ in the above equation, we use the expressions from Equations (6), (11), and (8). The first term in the numerator is the probability of correspondence of $\mathbf{p}_j$ with $\mathbf{s}_i$. This term is estimated as $\alpha_{ij}$ as in Equation (11). Incorporating the outliers $\epsilon$ in the model, the second term in the numerator is estimated as $(1-\epsilon)p(\mathbf{p}_j|\mathbf{s}_i)$. The denominator is obtained from Equation (8). Hence the expression for $p^{\text{old}}$ is written as,

$$p^{\text{old}}(\mathbf{s}_i|\mathbf{p}_j) = \frac{\alpha_{ij}\ \exp\left(-\frac{\|\mathbf{p}_j-\mathbf{s}_i\|^2}{2\sigma^2}\right)}{\frac{\epsilon(2\pi\sigma^2)^{d/2}}{n(1-\epsilon)} + \sum_{k=1}^{m}\alpha_{ik}\exp\left(-\frac{\|\mathbf{p}_j-\mathbf{s}_k\|^2}{2\sigma^2}\right)}. \quad (14)$$

*M-step*: In the M-step, *"new"* parameter set is obtained by minimizing the expectation of the complete negative log-likelihood function of Equation (12) as,

$$\mathcal{L} = -\sum_{j=1}^{n}\sum_{i=1}^{m+1} p^{\text{old}}(\mathbf{s}_i|\mathbf{p}_j)\log(p^{\text{new}}(\mathbf{s}_i)p^{\text{new}}(\mathbf{p}_j|\mathbf{s}_i)). \quad (15)$$

Ignoring the constants independent of $\Theta$, we can write the expression for complete likelihood $\mathcal{L}_c$ as (Ma et al., 2016),

$$\mathcal{L}_c = \frac{1}{2\sigma^2}\sum_{j=1}^{n}\sum_{i=1}^{m+1} p^{\text{old}}(\mathbf{s}_i|\mathbf{p}_j)\,\|\mathbf{p}_j-\mathbf{s}_i\|^2 + \frac{\xi d}{2}\log(\sigma^2) - \xi\log(1-\epsilon) - (m-\xi)\log(\epsilon) + \lambda\,||\mathbb{G}||, \quad (16)$$

where $\xi = \sum_{j=0}^{n}\sum_{i=0}^{m} p^{\text{old}}(\mathbf{s}_i|\mathbf{p}_j)$, and the last term is a regularizer weighted by a factor $\lambda$ to restrict the spread of the data by the transformation. For a skeleton point $\mathbf{s}_i$ and the corresponding transformed point $\mathbf{s}'_i$ is regularized as the Gaussian kernel, $\mathbb{G}(\mathbf{s}_i,\mathbf{s}'_i) = \exp -||\mathbf{s}_i - \mathbf{s}'_i||^2/2\beta^2$, where the variable $\beta$ controls the spread of the data. If the value of $\beta$ is too small, the transformation will have negligible effect on the data, while large value of $\beta$ will result in badly scattered points. Minimizing $\mathcal{L}_c$ decreases the negative log-likelihood of the energy function.

The EM algorithm iterates until there is not sufficient update of the parameters (we keep the tolerance at $10^{-4}$). The updated location of skeleton point set $\mathbb{S}$ is the optimal transformed point set $\mathbb{S}^*$.

## 4. EXPERIMENTS AND ANALYSES

We performed several experiments with real world and synthetic datasets in both qualitative and quantitative manners. More specifically, we have experimented with the data from 3 datasets. The first dataset we have used is the PlantScan3D dataset (pla, 2010), which contains different varieties of plants including Cherry, and Apple tree point cloud data. Next, we used the dataset from Tabb and Medeiros (2018). The dataset contains "Fuji" apple tree data in voxel format. We have extracted the raw point cloud from the data, and performed experiments without using any connectivity information from the voxel structure. Finally, we have generated point cloud data by scanning real *Arabidopsis thaliana* plants. For all the datasets, we obtained the initial coarse skeleton using the method of Xu et al. (2007), and performed the proposed optimization technique. Although other types of skeletonization methods can be used to obtain the initial skeleton, the result of the optimization algorithm depends on the quality of the initial skeleton. We have performed the optimization by using Space Colonization algorithm (Runions et al., 2007; Preuksakarn et al., 2010) as the initial skeleton, where we selected the set of parameters which produced best results. We noticed that in many cases, the results suffer from problems where the skeleton points are too far from the centerline. Especially the branch junction points are located out of the boundary of the point cloud in many cases. The optimization performs poorly in these cases than using the method of Xu et al. (2007) as the initial skeleton. We used Python 2.7 version along with PlantScan3D library for the implementation. The experiments were performed in a MacBook Pro with 2.2 GHz Intel Core i7 processor and 32GB DDR4 RAM. The datasets in **Figure 4** contain about 124, 30, and 18 k points in the Apple tree, Cherry tree, and Arabidopsis data, respectively. The respective computation times are 4080 s, 81 s, and 34 s.

First we demonstrate some qualitative results in **Figure 3**, where we show the skeletonization results on Arabidopsis plant and Cherry tree point cloud data. For visual clarity, we show part of the original data for the Arabidopsis plant. Similarly for the Cherry tree data, we removed the noise and truncated part of a branch to demonstrate the quality of the skeleton. In **Figure 4**, we show the full skeleton structures along with the original point cloud data for a "Fuji" apple tree (Tabb and Medeiros, 2018), cherry tree, and arabidopsis plant, respectively.

Next, we demonstrate the quantitative results of different plant phenotypes obtained from the computed skeleton. We have generated ground truth of junction points of the branches (branching point) for 3 cases: synthetic data, Arabidopsis plant data, and Cherry tree data. We notice that the quality of the skeleton explicitly affects the location of the branching point and the relative length of branch segment between the branching points with respect to the average branch length. By branch point, we mean the point where two or more branches meet. By branch segment, we mean the length/distance between two consecutive branch points. In order to evaluate the quality of the skeleton, we use two metrics based on the above mentioned factors to quantify the errors. The first metric is the deviation of the computed junction point from the ground truth location, and the second metric is the difference between the ground truth branch segment length and the computed branch segment length with respect to the average branch length. We compare our results with the state-of-the-art skeletonization algorithms which are built specifically for plants, consider raw point cloud data as input to the algorithm, and the implementation is publicly available. More specifically, we compare our algorithm with Xu et al. (2007) and Space Colonization algorithm (Runions et al., 2007; Preuksakarn et al., 2010) implemented in PlantScan3D library. In **Figure 5**, we show the error bar plots for relative distance error of junction point location and branch segment length error in top and bottom row, respectively.

We also show the statistical distribution of the corresponding error values in the box plots of **Figure 6**. Similar to **Figure 5**, the top row shows the results of junction point location errors

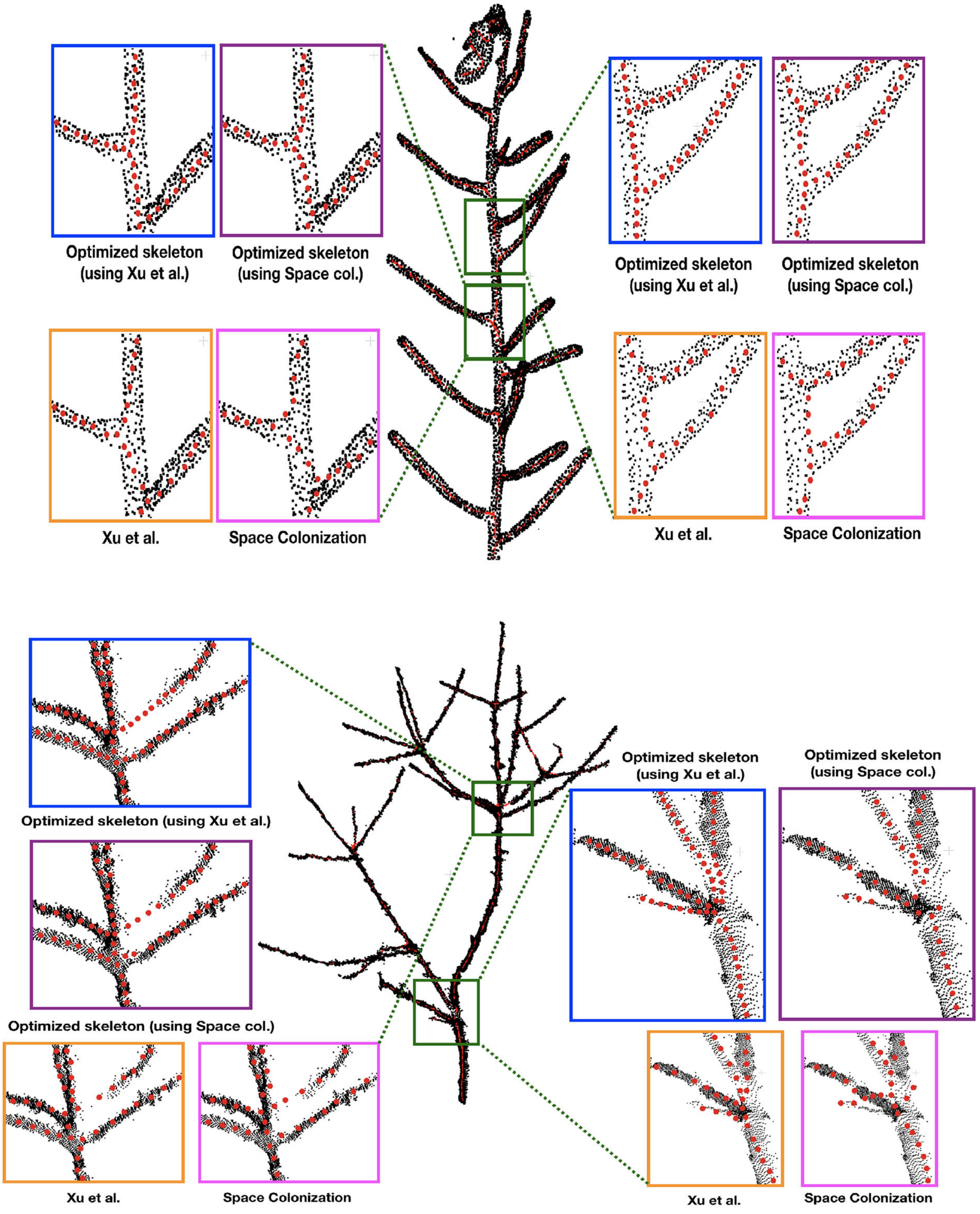
Optimized skeleton (using Xu et al.)
Optimized skeleton (using Space col.)
Optimized skeleton (using Xu et al.)
Optimized skeleton (using Space col.)
Xu et al.
Space Colonization
Xu et al.
Space Colonization
Optimized skeleton (using Xu et al.)
Optimized skeleton (using Space col.)
Optimized skeleton (using Xu et al.)
Optimized skeleton (using Space col.)
Xu et al.
Space Colonization
Xu et al.
Space Colonization

**FIGURE 3 |** Skeletonization results on 2 datasets: Arabidopsis plant **(top)** and Cherry tree **(bottom)** point cloud data. Note that the red skeleton points lie inside the plant stem, and that is why the visibility is not always clear due to the occlusion by the original (black) point cloud. We show some parts of the data by zooming-in and showing the skeleton points at the surface of the plant by taking the points out of the plant.

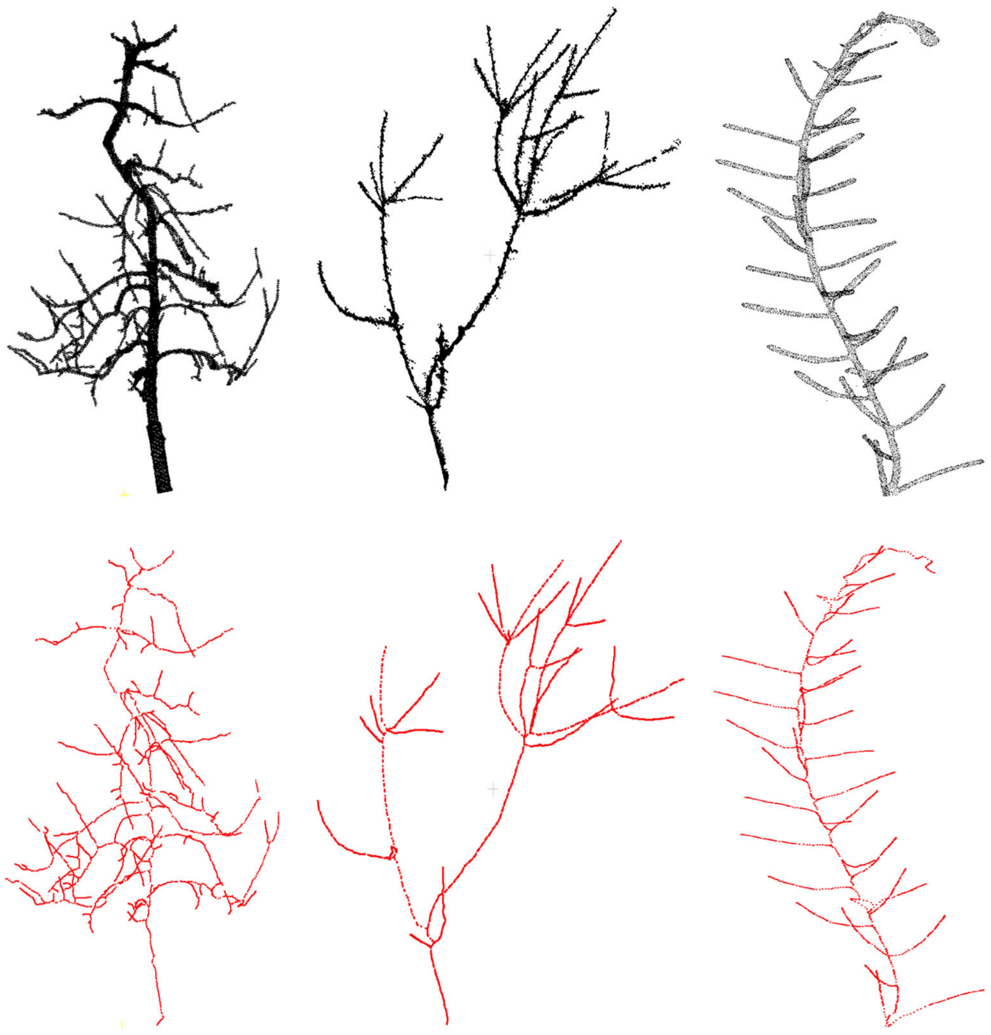

**FIGURE 4 |** Results of skeletonization on the whole plant point cloud data. **(Left)** Dataset from Tabb and Medeiros (2018), **(middle)** apple tree dataset from INRIA Plantscan3d dataset, **(right)** arabidopsis data.

and the bottom row shows the results of relative branch segment length errors. As can be seen from the figure, the span of error values for the proposed optimization algorithm is always less than the other methods.

## 5. DISCUSSION

In this work, we have proposed a technique to improve an existing skeleton by a stochastic optimization technique. The idea of exploiting the original point cloud data in the optimization process allows the skeleton to retain biological relevance with respect to the input point cloud data. The method is tested on different types of data, and improvement over the existing technique is demonstrated. We have demonstrated the effectiveness of the approach by quantifying the junction point location and branch length segment errors, where the branch segment is considered as the length between two consecutive junction points. While the performance varies among the type of the datasets, the optimization clearly improves over the initial skeleton obtained by state-of-the-art (Runions et al., 2007; Xu et al., 2007; Preuksakarn et al., 2010) in the general case. The proposed optimization method can be beneficial to accurate measurements of different types of plant phenotypes.

The method is designed for plants of moderate size having about 100 k points. To optimize skeleton of large tree with huge number of points, computation time will be a crucial factor. In these cases, the number of points need to be downsampled to obtain the optimization result in reasonable amount of time. In this work, we did not consider optimizing the computation time or parallelizing the solution for multiple processors. In the current framework, we are not considering the cases of plants with leaves. Although the method might be able to handle the type of leaves having long and narrow shape, but handling other

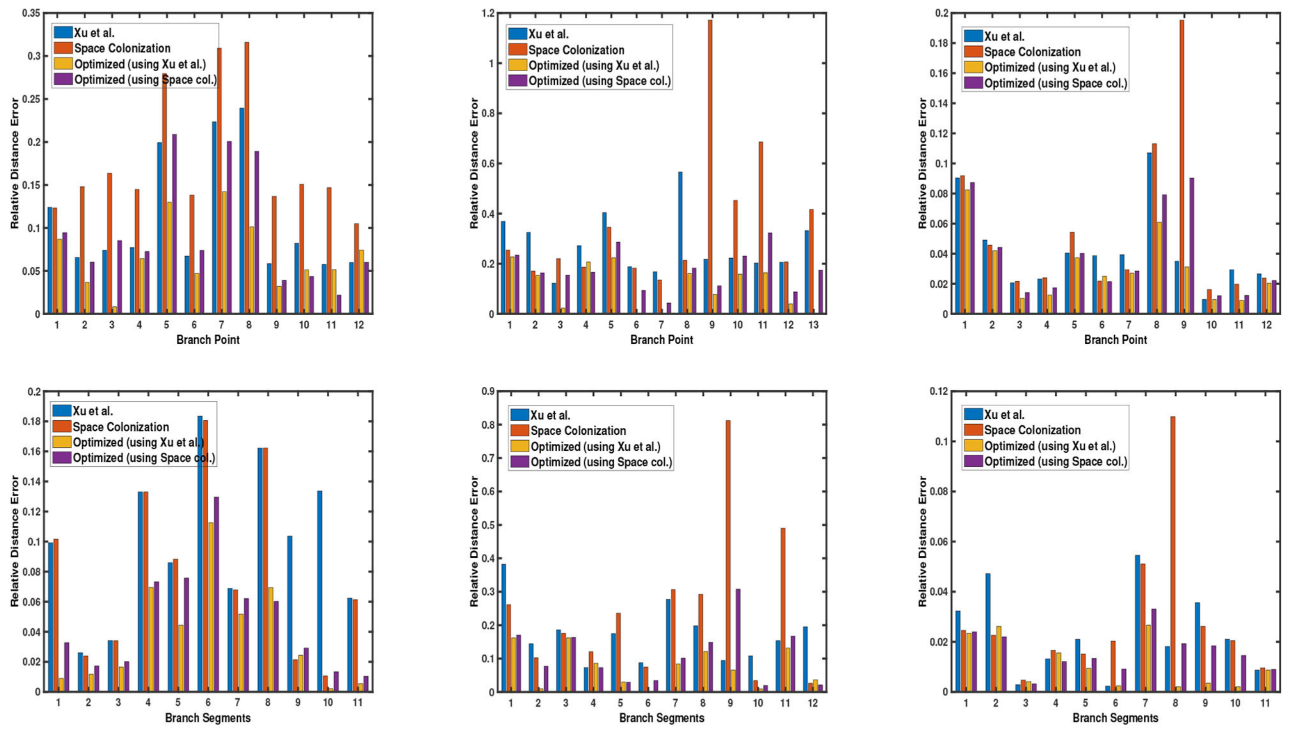

**FIGURE 5 |** Error bar graphs for relative junction point location **(top)** and branch segment length **(bottom)** for 3 datasets: a synthetic data, Arabidopsis plant data (shown in each column), and Cherry tree data. In each graph, the vertical bar represents the value of error with respect to the groundtruth. Performance of each algorithm is shown in different colors, as defined by the legend in each graph.

types of complex leaf shapes will be challenging. Given the large variety of leaves of different types of plant species, considering the general case of any type of leaf is beyond the scope of the current work. Another crucial factor is the initial skeleton, which is used as the starting point of the optimization. We have demonstrated that using the method of Xu et al. (2007) can be reliably used as the initial skeleton for varieties of cases. However, other types of methods can also be used to compute the primary skeleton, given the fact that the skeleton is reasonably good. As shown in the results, optimization using the Space Colonization (Runions et al., 2007; Preuksakarn et al., 2010) as the initial skeleton is also performed. However, the results are slightly worse than using the method of Xu et al. The reason is that, the optimization result is dependent on the starting point. If the initial skeleton points are too far from the centerline (which results in many cases of Space Colonization algorithm), the optimization fails to move the points to the exact center. Basically the initial skeleton points which are far away, results in getting the optimization to stuck to a bad local minima. In the current formulation we did not consider handling these type of cases, and we assume that the initial skeleton is reasonably correct. The optimization parameters $\lambda$ and $\beta$ play an important role on the final result. By default we keep the values as $\lambda = 5$, $\beta = 5$. We tested with different values of $\lambda$ and $\beta$ between the range 1 to 15, and in general the best results are obtained with the default values as stated above. However, we notice that for larger trees, higher values of $\lambda$ and $\beta$ slightly improve the result. For any particular species, using a fixed set of values of the parameters should be sufficient.

The centerline assumption of the Gaussian Mixture Model is tested on small plants. However, we have not tested the optimization performance for very thick stems, where the skeleton curve might have discontinuous derivatives or high curvature near the branching points. In order to reconstruct the geometry of the branching structure by the notion of generalized cylinders, it will be problematic with the high curvature areas. Handling these types of cases are not considered here, and the framework might need to be reformulated for considering more complex and thicker branching structures. With the goal of reconstructing the geometry of the original plant point cloud data using the skeleton, finding a minimum spanning tree that optimizes the connection between the discrete skeleton points in terms of some biological prior might further improve the skeleton structure. However, this is a challenging problem, and we leave it as a future work. Similar type of problem is handled by some heuristic approach in Wu et al. (2019), but a generic solution to the problem can be a major part of plant geometry reconstruction.

The proposed method considers plants having no leaves. Skeletonization by considering plant leaves will be worth studying in order to reconstruct the geometry of leafy plants. We have not considered the case where there is lot of noise in

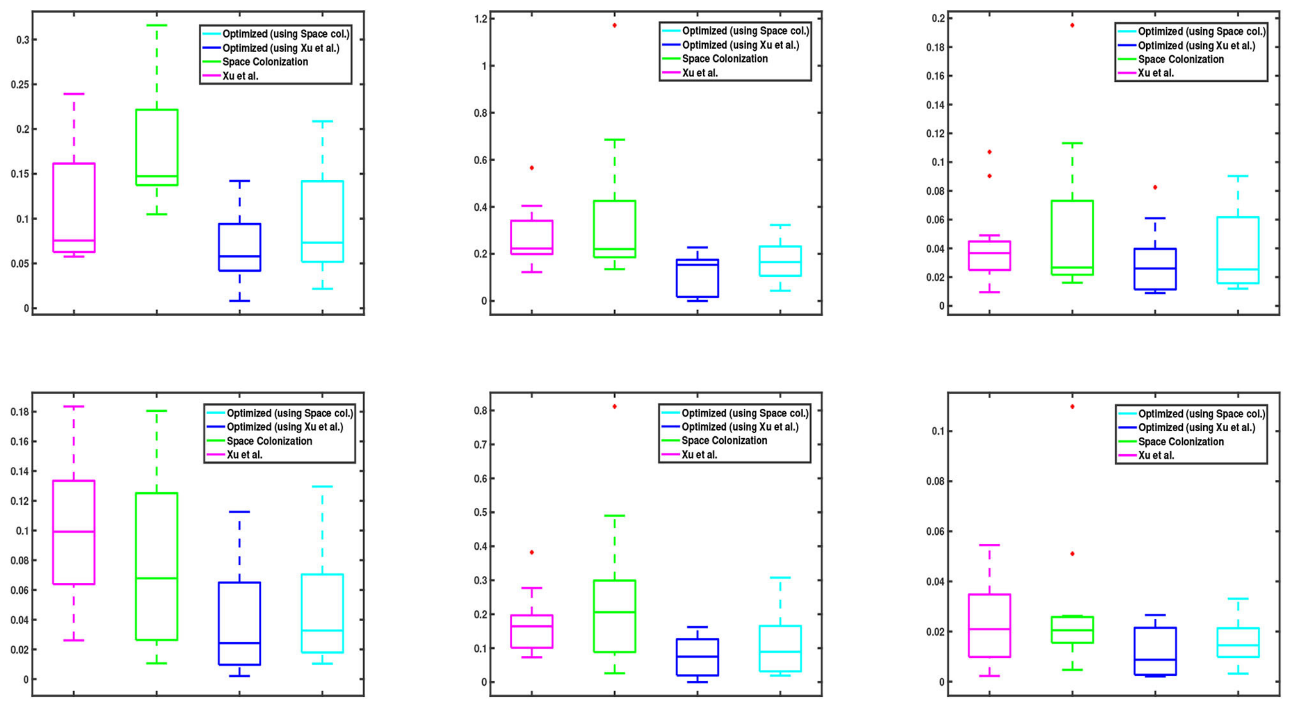

**FIGURE 6 |** Box plots corresponding to the data values in **Figure 5**. **(Top and bottom)** row shows the relative results of junction point location and branch segment length errors with respect to the groundtruth, respectively. In each graph, the central mark of each box indicates the median, while the bottom and top edges of the box indicate the 25-th and 75-th percentiles of the data, respectively. The whiskers extend to the most extreme data points which are not considered as outliers. The red "+" symbols plot the outliers in the data, if there is any.

the data. Also, explicit handling of occlusion is not modeled in the framework.

## DATA AVAILABILITY STATEMENT

The datasets generated for this study are available on request to the corresponding author.

## AUTHOR CONTRIBUTIONS

AC and CG designed the study, developed the technical ideas, and wrote and approved the manuscript together. AC implemented the code, generated the figures, and graphs.

## FUNDING

This work was supported by Robotics for Microfarms (ROMI) European project.

## ACKNOWLEDGMENTS

We would like to thank Frédéric Boudon for useful discussions and making the implementation of PlantScan3D library available for public use. We would also like to thank Julie Charlaix for collecting and sharing the Arabidopsis data, and other members of the ROMI project for the (not yet published) pipeline to reconstruct the point cloud from images.

## REFERENCES

(2010). *PlantScan3D*. Available online at: https://gforge.inria.fr/frs/?group_id=2512 (accessed January 2020).

Brichet, N., Fournier, C., Turc, O., Strauss, O., Artzet, S., Pradal, C., et al. (2017). A robot-assisted imaging pipeline for tracking the growths of maize ear and silks in a high-throughput phenotyping platform. *Plant Methods* 13:96. doi: 10.1186/s13007-017-0246-7

Bucksch, A., Lindenbergh, R. C., and Menenti, M. (2009). "Skeltre - fast skeletonisation for imperfect point cloud data of botanic trees," in *Proceedings of Eurographics Conference on 3D Object Retrieval* (Munich), 13–20.

Burger, P., and Gillies, D. (1989). *Interactive Computer Graphics*. Addison-Wesley Publishing Company.

Cao, J., Tagliasacchi, A., Olson, M., Zhang, H., and Su, Z. (2010). "Point cloud skeletons via laplacian based contraction," in *Proceedings of Shape Modeling International (SMI)* (Aix-en-Provence), 187–197.

Chattopadhyay, S., Akbar, S. A., Elfiky, N. M., Medeiros, H., and Kak, A. (2016). "Measuring and modeling apple trees using time-of-flight data for automation of dormant pruning applications," in *IEEE Winter Conference on Applications of Computer Vision (WACV)* Lake Placid, NY, 1–9.

Chaudhury, A., Ward, C., Talasaz, A., Ivanov, A. G., Brophy, M., Grodzinski, B., et al. (2019). Machine vision system for 3d plant phenotyping.

*IEEE/ACM Trans. Comput. Biol. Bioinformatics* 16, 2009–2022. doi: 10.1109/TCBB.2018.2824814

Choudhury, S. D., Samal, A., and Awada, T. (2019). Leveraging image analysis for high-throughput plant phenotyping. *Front. Plant Sci.* 10:508. doi: 10.3389/fpls.2019.00508

Cornea, N. D., Silver, D., and Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Trans. Visual. Comput. Graph.* 13, 530–548. doi: 10.1109/TVCG.2007.1002

Delagrange, S., Jauvin, C., and Rochon, P. (2014). Pypetree: a tool for reconstructing tree perennial tissues from point clouds. *Sensors* 14, 4271–4289. doi: 10.3390/s140304271

Godin, C., and Caraglio, Y. (1998). A multiscale model of plant topological structures. *J. Theor. Biol.* 191, 1–46. doi: 10.1006/jtbi.1997.0561

Goodman, T. N. T. (1985). Properties of beta splines. *J. Approx. Theory* 44, 132–153. doi: 10.1016/0021-9045(85)90076-0

Goodman, T. N. T., and Unsworth, K. (1986). Manipulating shape and producing geometuic contnuity in beta spline curves. *IEEE Comput. Graph. Appl.* 6, 50–56. doi: 10.1109/MCG.1986.276692

Hackenberg, J., Morhart, C., Sheppard, J., Spiecker, H., and Disney, M. (2014). Highly accurate tree models derived from terrestrial laser scan data: a method description. *Forests* 5, 1069–1105. doi: 10.3390/f5051069

Huang, H., Wu, S., Cohen-Or, D., Gong, M., Zhang, H., Li, G., et al. (2013). L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32, 65:1–65:8. doi: 10.1145/2461912.2461913

Jian, B., and Vemuri, B. C. (2011). Robust point set registration using gaussian mixture models. *IEEE Trans. Pattern Anal. Mach. Intell.* 33, 1633–1645. doi: 10.1109/TPAMI.2010.223

Jin, D., Iyer, K. S., Chen, C., Hoffman, E. A., and Saha, P. K. (2016). A robust and efficient curve skeletonization algorithm for tree-like objects using minimum cost paths. *Pattern Recogn. Lett.* 76, 32–40. doi: 10.1016/j.patrec.2015.04.002

Liang, X., Kankare, V., Yu, X., Hyyppä, J., and Holopainen, M. (2013). Automated stem curve measurement using terrestrial laser scanning. *IEEE Trans. Geosci. Remote Sens.* 52, 1739–1748. doi: 10.1109/TGRS.2013.2253783

Livny, Y., Yan, F., Olson, M., Chen, B., Zhang, H., and El-Sana, J. (2010). Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. Graph.* 29, 151:1–151:8. doi: 10.1145/1882261.1866177

Ma, J., Zhao, J., and Yuille, A. L. (2016). Non-rigid point set registration by preserving global and local structures. *IEEE Trans. Image Process.* 25, 53–64. doi: 10.1109/TIP.2015.2467217

Myronenko, A., and Song, X. B. (2010). Point set registration: coherent point drift. *IEEE Trans. Pattern Anal. Mach. Intell.* 32, 2262–2275. doi: 10.1109/TPAMI.2010.46

Preuksakarn, C., Boudon, F., Ferraro, P., Durand, J. B., Nikinmaa, E., and Godin, C. (2010). "Reconstructing plant architecture from 3d laser scanner data," in *Proceedings of 6th International Workshop on Functional Structural Plant Models* (Davis, CA).

Prusinkiewicz, P., and Lindenmayer, A. (1990). *The Algorithmic Beauty of Plants*. New York, NY: Springer-Verlag.

Raumonen, P., Kaasalainen, M., Åkerblom, M., Kaasalainen, S., Kaartinen, H., Vastaranta, M., et al. (2013). Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sens.* 5, 491–520. doi: 10.3390/rs5020491

Rodkaew, Y., Chongstitvatana, P., Siripant, S., and Lursinsap, C. (2003). "Particle systems for plant modeling," in *Proceedings of Plant Growth Modeling and Applications* (Beijing), 10–17.

Runions, A., Lane, B., and Prusinkiewicz, P. (2007). "Modeling trees with a space colonization algorithm," in *Proceedings of Eurographics Workshop on Natural Phenomena* (Prague), 63–70.

Spalding, E. P., and Miller, N. D. (2013). Image analysis is driving a renaissance in growth measurement. *Curr. Opin. Plant Biol.* 16, 100–104. doi: 10.1016/j.pbi.2013.01.001

Tabb, A., and Medeiros, H. (2018). "Fast and robust curve skeletonization for real-world elongated objects," in *IEEE Winter Conference on Applications of Computer Vision* (Lake Tahoe, NV).

Tagliasacchi, A., Delame, T., Spagnuolo, M., Amenta, N., and Telea, A. (2016). "3D skeletons: A state-of-the-art report," in *Proceedings of the 37th Annual Conference of the European Association for Computer Graphics: State of the Art Reports* (Lisbon), 573–597.

Tagliasacchi, A., Zhang, H., and Cohen-Or, D. (2009). Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph.* 28, 71:1–71:9. doi: 10.1145/1531326.1531377

Tardieu, F., Cabrera-Bosquet, L., Pridmore, T., and Bennett, M. (2017). Plant phenomics, from sensors to knowledge. *Curr. Biol.* 27, R770–R783. doi: 10.1016/j.cub.2017.05.055

Vázquez-Arellano, M., Griepentrog, H. W., Reiser, D., and Paraforos, D. S. (2016). 3-D imaging systems for agricultural applications–a review. *Sensors* 16, 1–24. doi: 10.3390/s16050618

Wu, S., Wen, W., Xiao, B., Guo, X., Du, J., and Wang, C. (2019). An accurate skeleton extraction approach from 3D point clouds of maize plants. *Front. Plant Sci.* 10:248. doi: 10.3389/fpls.2019.00248

Xu, H., Gossett, N., and Chen, B. (2007). Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. Graph.* 26, 19:1–19:13. doi: 10.1145/1289603.1289610

Yan, D., J, Wintz, Mourrain, B., Wang, W., Boudon, F., and Godin, C. (2009). "Efficient and robust reconstruction of botanical branching structure from laser scanned points," in *IEEE International Conference on Computer Aided Design and Computer Graphics (CAD)* (Huangshan).

Zeng, G., Wang, J., Kang, S. B., and Quan, L. (2007). Image-based tree modeling. *ACM Trans. Graph.* 26, 87:1–87:7. doi: 10.1145/1276377.1276486

Zhang, D., Xie, N., Liang, S., and Jia, J. (2015). 3D tree skeletonization from multiple images based on pyrlk optical flow. *Pattern Recogn. Lett.* 76, 49–58. doi: 10.1016/j.patrec.2015.11.007

Zhang, X., Huang, C., Wu, D., Qiao, F., Li, W., Duan, L., et al. (2017). High-throughput phenotyping and QTL mapping reveals the genetic architecture of maize plant growth. *Plant Physiol.* 173, 1554–1564. doi: 10.1104/pp.16.01516

Zhen, W., Zhang, L., Fang, T., Tong, X., Mathiopoulos, P. T., Zhang, L., et al. (2016). A local structure and direction-aware optimization approach for three-dimensional tree modeling. *IEEE Trans. Geosci. Remote Sens.* 54, 4749–4757. doi: 10.1109/TGRS.2016.2551286

Ziamtsov, I., and Navlakha, S. (2019). Machine learning approaches to improve three basic plant phenotyping tasks using three-dimensional point clouds. *Plant Physiol.* 181, 1425–1440. doi: 10.1104/pp.19.00524

**Conflict of Interest:** The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.