Geometry Reconstruction of Plants

Ayan Chaudhury, Christophe Godin

Geometry Reconstruction of Plants

Ayan Chaudhury and Christophe Godin

INRIA Grenoble Rhône-Alpes, team MOSAIC
Laboratoire Reproduction et Développement des Plantes, Univ Lyon
ENS de Lyon, UCB Lyon 1, CNRS, INRA
46 allée d'Itale, 69364 Lyon Cedex 07, France
{ayan.chaudhury, christophe.godin}@inria.fr

Abstract. Synthetic modelling and reconstruction of the geometry of real plants have been a center of attention of research for decades. Due to the complex architecture and growth pattern of plants, accurate modelling of the plant geometry is an extremely challenging task. Although realistic modelling of plants are widely studied in the context of computer graphics research, it also has profound impact on the biological study of plants. In order to perform various types of simulation studies under different environmental conditions and in understanding the physiology of plants in more details, synthetic models can be an extremely useful tool. Synthetic modelling approaches can be broadly categorized into three types. The first type is the rule based procedural modeling approach, which does not account the real data into consideration. The second type of approach (also known as *data driven modelling*) performs modelling based on the real data obtained from 3D acquisition procedures. The third type of approach is interactive, which is based on user assistance. In this chapter, we focus on the modelling of the second category and revisit the recent state-of-the-art techniques performing reconstruction of plant geometry from real data. The algorithms can be classified into different interlinked categories, which constitute the general pipeline of geometry reconstruction in data driven modelling framework. In the context of biological relevance of different types of techniques, we discuss about the strengths and limitations of the approaches and the need of *prior* botanical knowledge to reconstruct the plant geometry in biologically feasible manner. Finally, we explore the quantitative assessment techniques which can be used to measure the quality of the reconstruction result with respect to the actual data.

Keywords: Geometry Modelling, Reconstruction, Point Cloud, Skeleton, Phenotyping, Laser Scan, Data Driven Modelling, Flower Reconstruction, Quantitative Evaluation.

1 Introduction

Modelling the interesting and complex geometry of plants have been a center of attention of research for biologists and mathematicians for decades [1, 7]. Numerous approaches have been proposed in order to mathematically model the

geometrical structure of plants in a robust manner. While one motivation of studying the plant geometry is to better understand the structure of plants from mathematical perspective, realistic modelling of plants have become an active area of research in computer graphics nowadays, in particular due to the explosion of phenotyping platforms for agronomic research [8]. We are fast progressing towards intelligent imaging based systems for automated analysis of different types of phenotypic traits of plants. Due to the non-invasive nature and automated analysis techniques, imaging based systems are getting extremely popular. Now we start to be able to reconstruct a full 3D model of a plant from multiple view images which mimics the original plant, and can extract desired biological features of the plant in a non-invasive way [9]. Numerous systems have been proposed to study the effect of different environmental conditions on various types of plant species as well as on the genetically modified versions. These types of systems are extremely useful in order to quantitatively analyze different experimental outcomes in the field of crop science research. For example, one might want to study the effect of different types of lighting conditions on different genetic modifications of a particular species, the growth pattern of a plant over certain period, or track a leaf development in order to study the shape changes over time. With the advent of imaging based technologies, such analyses are becoming possible in a fully automated manner.

Although the notion of *realistic* plant model in computer graphics and animation does not necessarily imply the actual biological relevance of the models with the *real* plants, the ultimate goal of synthetic plant modelling is to mimic the apperance of the original plant, both in terms of geometrical structure and of the texture realism effect. Synthetic modelling of plants can be beneficial to broadly two types of research studies. First, this can be a quintessential tool to perform simulation experiments. Accurate modelling of the plant geometry can help us to build models which can be used to simulate the behavior of the actual plant [10]. For example, let's say we intend to study the mechanical effect of wind on a particular plant. If we have reconstructed the 3D model of a plant in such a way that the mathematical representation and interaction of different organs are well defined in terms of plant dynamics, we can simulate the effect of wind on different organs and simultaneously can model the behavior of the plant as a whole in biologically plausible manner. Similarly, we might be able to simulate the growth of a plant over time from its synthetic model. The goal is to study specific aspects of the plant behavior in a fully virtual manner without the need of experimentation with a real plant. One can compare this with a flight simulator, where the actual flying is virtually simulated without the expensive process of real flying. Similar type of modelling of chemical bonding structures are also studied in drug development research. Basically the goal is to minimize the overhead and luxury of working on a real object, and replace it with a synthetic model.

Second type of application of synthetic models include quantitative analysis of different types of biological traits of plants. The models can help us to compute phenotypes such as internode distances, insertion and divergence an-

gles of individual branches, length and diameter of each branch, extraction of branching points, etc. in a fully automatic manner. Also, studying properties like plant growth (e.g. volume, surface area), change of leaf shape, inflorescence of flowers, etc. is possible with the help of synthetic models. However, one of the biggest challenges associated with these type of analyses is to accurately reconstruct the geometry of the plant under consideration. Due to the complex structure of plants, the problem is extremely challenging. For example, self occlusion results in incomplete or missing data problem, which directly affects the quality of geometry reconstruction in occluded regions. Also, reconstruction of very thin branches is a common problem in this context. Various types of computer vision and machine learning algorithms are reported in the literature to solve these type of problems, but most of the techniques are task specific due to the lack of robustness and lack of generality to be applied to varieties of plants. In general, geometry modelling of plants can be broadly categorized into three types of approaches [29]. The first type of approach is the rule based procedural modelling approach. This type of approach is based on a set of predefined rules to define complex objects by successively replacing parts of a simple initial object. A classical example of this type of approach is the L-system [1]. Although rule based procedural approaches have been widely used in plant modelling, it is hard to incorporate the framework to work on real data and reconstruct the original plant geometry. Also, the implementation of procedural model is very cumbersome. However, due to the recent advancements of the implementation framework of L-system based modelling using high level programming languages, the approach has become more flexible to implement [2, 31]. The second type of modelling approach is based on real data (e.g. 2D image or 3D point cloud format), called *data driven* modelling. The goal of this type of approach is mostly focused on the biological relevance of the model to reconstruct the geometry of the original plant from the input data. Finally, the third type of approach is based on interaction from the user, and may be used to refine the result of previous two approaches [3]. This approach is mostly used in computer graphics and animation applications, where the user can interactively edit the geometry according to the need of the aplication.

In this chapter, we focus on the state-of-the-art techniques to reconstruct the plant geometry from real data (*data driven modelling* techniques). A large number of works are reported in the literature focusing on different aspects of data driven modelling techniques. We observe that these techniques can be clustered into different categories. Each of these categories constitute a block of a pipeline, which can be thought of as a general sequence of steps for data driven modelling. In a broader sense, the pipeline consists of the following main building blocks: 1) Raw point cloud extraction, 2) Connectivity graph formation, 3) Initial skeleton construction, 4) Skeleton refinement, 5) Organ surface and volume reconstruction. The pipeline can be visually represented as in Figure 1, where each circle represents a building block (or a *state*). In general, many of the techniques follow all the steps in the pipeline in a sequential manner, and some algorithms skip certain steps. Basically the underlying idea is that,

the pipeline can be followed only in the forward direction, and skipping states is allowed. The possible transitions are indicated by the arrows in the figure. The states are sequentially numbered at the bottom, and we use the following convention to denote the steps followed by a particular algorithm. For example, if an algorithm takes raw point cloud data as input, constructs a graph from the data, followed by skeletonization of the input data using the graph, and finally reconstructs a particular organ of the plant, we denote the transition sequence as, $1 \rightarrow 2 \rightarrow 3 \rightarrow 5$. In the next section we discuss about the building blocks of the pipeline with respect to the recent state-of-the-art techniques.
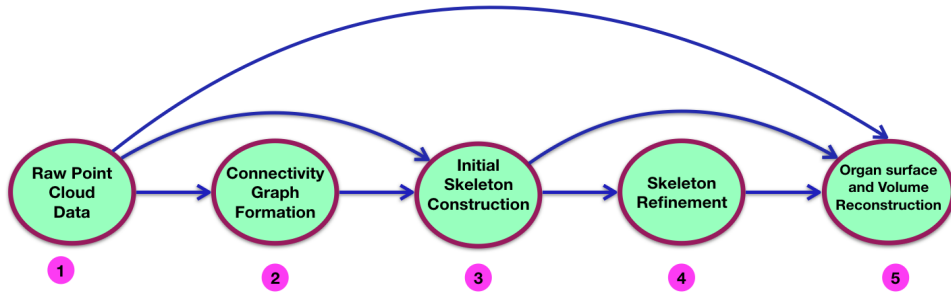
Fig. 1: The general pipeline of data driven modelling. Each circle represents a block of the pipeline and the arrows represent possible transitions of the work-flow.

2   Data Driven Modelling of Plants

2.1   Step-1 : Raw Point Cloud Data Extraction

A point cloud is a set of data points, which are the 3D point coordinates at the surface of an object. Point clouds can be generated by different types of techniques. One of the primitive techniques for point cloud generation was the manual *digitization strategy by contact* [4], where the user-clicks on the surface of the object (plant) are recorded and stored as point cloud format. In recent years, two main techniques are widely used in generating point cloud data: back projection technique to generate 3D data from a sequence of overlapping 2D images around the plant ([5, 6]), and laser scanner technology to record surface point coordinates at the surface of an object ([32–34]). While 2D image based techniques generate 3D data using *structure from motion* techniques, laser scanning is an efficient way to obtain 3D data directly by scanning. Laser scanning is a state-of-the-art technology to perform non-invasive 3D analysis of plant phenotyping. Depending on the need of the application, the resolution of the point

spacing in the generated point cloud is possible to control. The 3D model of the plant can be generated by registering multiple overlapping views of overlapping scans.

2.2   Step-2 : Connectivity Graph Formation

The raw point cloud data does not contain any topological connectivity information of the points. One of the very first step of geometry reconstruction consists of constructing a graph from the raw point cloud data. A typical way to construct the graph is to connect the points by nearest neighbour strategy (e.g. *kd*-tree search). The graph is also known as Riemannian graph. The points represent vertices of the graph, and the edges are denoted by connection of the vertices. The edge weight is determined by the distance between the vertices connected by the edge. Different types of distance measures can be used (e.g. Euclidean distance, geodesic distance, etc). Graphs having edge weights are called wighted Riemannian graph. Riemannian graph structure is used as a basis of geometry reconstruction by many techniques [13, 14, 18].

2.3   Step-3 : Initial Skeleton Construction

A skeleton is a thin structure obtained from an object shape, which encodes the topology and basic geometry of the original object. Skeletons are compact representation of the shape and is extremely useful in various types of shape analysis tasks [11]. Say $\mathcal{P} \in \mathbb{R}^3$ is a point set. Let's define a ball of radius r centered at $p \in \mathcal{P}$ as $\mathcal{S}(p) = \{q \in \mathbb{R}^3, d(p,q) < r\}$, where $d(p,q)$ is the distance between p and q in $\mathbb{R}^3$. A maximal ball is defined as a ball $\mathcal{S}(p) \subset \mathcal{P}$ which is not completely included in any other ball included in $\mathcal{P}$. A general way of defining skeleton is that, a skeleton is the set of centers of all maximal balls included in $\mathcal{P}$. Extracting accurate skeleton from a shape is an extremely challenging task. By *accurate*, it means that ideally the skeleton should pass through the exact centerline of the shape. In other words, within a local neighbourhood of the shape, the distances from the skeleton point to the enclosing shape boundaries should be the same. Although 2D skeleton extraction is a well studied problem in the literature, 3D skeletonization of point cloud data is still an open problem. The task is even more challenging in the case of plants, because of their complex geometry. Moreover, the problem of non-uniform point density and missing data make the task extremely complicated.

Different types of techniques have been proposed for skeleton extraction from point cloud data. Some of these approaches are based on Riemanian graph construction, and some of them compute skeleton from the point cloud data along with the sequence of 2D images. While these skeletons are constructed in the form of a graph, there might be loops in the graph due to the factors like close branches, missing data, etc. We discuss about recent advancements of skeletonization techniques along with their strengths and limitations.

A semi-automatic method of skeletonization by Xu *et al.* [13] is one of the notable work on skeleton extraction from point cloud data. The workflow transition sequence is $1 \rightarrow 2 \rightarrow 3$ as in Figure 1. The goal is to reconstruct the original plant model along with *completing* the missing information caused by self occlusion. Initially, assuming that the root of the tree is located at the bottom, all the points in the scanned data are connected as a weighted Riemanian graph, where the weight is defined as the edge length. Then the shortest path from each vertex in the graph to the root vertex are computed using Dijkstra's shortest path algorithm. The whole point cloud is then *clustered* based on the quantized shortest path lengths and graph adjacency information. The center of each cluster is assigned as a skeleton point (node), which are connected to each other according to their spatial locations. This idea produces a basic skeleton of the plant in the form of a graph. However due to occlusion and missing data, there might exist components (or sub-graphs) which are not connected to the main skeleton. In order to solve this problem, a breadth first traversal is initiated from the root node. During the traversal, a cone of certain angle is projected along the direction vector from the parent node to the current node. This process is performed for every node of the graph. If any point other than the nodes of the skeleton graph falls within the current volume of the cone, it is attached to the current node of the skeleton graph and becomes a node of the graph. Then the above mentioned process is repeated by considering newly added point as the root node and searching for more points closer to the current point. This ensures that disconnected parts are joined to the main skeleton in a meaningful way. This approach is reported to work well on deciduous trees (when the trees shed all the leaves) in a qualitative/visual evaluation manner. In order to extract the skeleton in the presence of leaves, some heuristics are used to synthesize the skeleton near the tree crown where most of the leaves are located. Finally, parametric curve smoothing (Hermite curve) is performed to incorporate "smooth" transition at the branching points to make the skeleton more realistic.

One problem associated with the technique is that, the skeleton points do not maintain the *centerdness* criteria, and results in a *zigzag* structure near the branching points (we discuss this issue later). This type of result does not support the biological relevance of the skeleton (although visually it might look good at a distance), and leads to incorrect geometry reconstruction by the data driven modelling pipeline. Also, the skeleton might contain *loops* resulting from close branches of the tree (this problem is later handled by Yan *et al.* [14] by transforming the skeleton graph into a tree structure using some heuristics).

Motivated by these type of problems, Bucksch *et al.* [22] proposed a skeletonization technique to compute tree parameters with the emphasis on botanical accuracy of measurements of the parameters (height, length and diameter of branches, etc). Instead of constructing Riemmanian graph from discrete points, the method exploits the voxel structure and applies some local heuristics to infer the connectivity of skeleton points (the state transition sequence is $1 \rightarrow 3$ in Figure 1). Initially the input point cloud is organized in an octree structure by subdividing the point cloud into octree cells, where each cell contains cer-

tain number of points. Then a graph is formed from the octree representation by forming vertices from the cells and connecting the vertices by edges using a nearest neighbour strategy. However, simply considering the center of gravity of each octree cell as graph vertex and joining the vertices by edges might lead to incorrect tree topology because of the inevitable noise and varying point density of typical real world laser scans. In order to solve this problem, a heuristic method is used. For any two adjacent octree cell centroids and the midpoint of the line connecting them, three planes are passed through these points, perpendicular to the line joining the points. The median values of the squared distances of the points in each cell to the corresponding planes (d_1 and d_2 respectively), and also the same for all the points to the intersecting plane is computed (d_3). Then based on a threshold on these values, the inference is made wheather the cells are connected to each other. The idea is that, if d_3 is sufficiently smaller than d_1 and d_2, then it is likely that the points in the cells are more "scattered" and there is a connection between the two cells (unlike the case where the points are "clustered" around a small neighbourhood in each cell, and thus it is likely that the cells are independent of each other). Finally, the centeredness of the skeleton is achieved by embedding the octree graph into the point cloud by a *graph embedding* technique [23]. Experimental results demonstrate the effectiveness of the technique on bad sampling conditions and accurate skeleton geometry reconstruction for various types of cases. However, the method is based on many heuristic assumptions and less robust to be useful for varieties of applications.

On a different type of approach, *particle flow* based techniques build the skeleton from the sequence of input photographs (state transition sequence $1 \rightarrow 3$ in Figure 1). The motivation of particle simulation based approach is based on the process of transport and exchange of energy, water and sustenance between roots, branches and leaves of a plant [15, 16].

Initially a standard segmentation algorithm is used to extract the tree from the background of the input photograph. Then some random (or user assisted) seed pixels are chosen, which are denoted as *particles*. By choosing the lowermost pixel at the root of the tree as target point, a path is traced starting from each seed pixel to the target. The path is created by standard *Livewire* segmentation technique [17] which is based on Dijkstra's shortest path algorithm using the edge information of the image. This idea is motivated by the fact that each particle is assumed to contain some energy, and the transportation rule directs each particle towards a target. The 2D skeleton or branching structure is computed by tracing the paths of the particles at the end of the simulation. Neubert *et al.* [16] called the combination of all the paths as *attractor graph* in an image. The final attractor graph is obtained by combining the direction vectors of the attractor graphs for each image plane. Next, a 3D voxel model of the tree is computed by assuming parallel projection model and back projecting the sequence of images (similar to the well known *space carving* approach). Each voxel is assigned to a density estimate and the 2D attractor graphs are used to compute the direction field of the particles (see [16] for details). These voxels and density information are used to initialize the positions of another set of particles (which are not the

same as the seed pixels used in the 2D image). Then the process of path tracing from each particle to the target point is repeated (as done before in the 2D image). However, a different strategy is used in the 3D case. Unlike the *Livewire* approach, the fundamental laws of Newtonian mechanics are used to compute the path. The position of a particle is updated according to certain rules of force and velocity of a particle [16], and nearby particles within some distance threshold are joined together to form a new particle. The trace of the particles forms the skeleton of the plant.

Although the technique is motivated by the energy transport phenomena of real plants, the skeleton geometry is not guaranteed to follow the actual geometry of the input data. The choise of seed points play an important role in forming the skeleton structure. Random placement of seed points may result in skeleton construction at wrong places. The attractor graph computation has inherent limitations. For the 2D attractor graph computation, *Livewire* technique is used to compute the branches of the tree. However, the generic *Livewire* technique is mainly designed to follow the edges of the image, but not spefically to follow the branching structures. For computing the 3D attractor graph, flow of the particles by the Newtonian mechanics rules are not supported by the branching geometry of the original plant.

Space colonization algorithm [19, 20] has been successful in modelling the complex branching structure of plants. The main idea of the algorithm is to generate a random set of point cloud at first to indicate the free space where the plant can grow. Then starting from the lowermost point (which will be the root of the reconstructed plant), *"eat-up"* the points (or particles) in the cloud in an iterative manner. It is assumed that each bud (undeveloped or embryonic shoot located at the tip of a stem) is surrounded by a spherical *occupancy zone* of certain radius (also called the *kill distance*) within a conical perception volume of certain angle and radius. Initially each particle is labelled as *unprocessed*. At every iteration, a bud searches in its perception volume to find the closest particles. If there are more than one particle in the volume, the branch direction is set to the average direction of all the particles from the current particle, and these particles are labelled as *processed*. The same operation is continued in every iteration, until there is no particle left in the cloud which is labelled as *unprocessed*. Although this idea can produce visually pleasing skeletal structures, it is not fully supported by the biological constraints of plants. The *"eat-up"* process might result in creation of branching structure in the wrong direction, and there is no mechanism to constrain the process to the original branching structure. Space colonization algorithm was proposed to generate synthetic plants, and no real dataset was used to reconstruct the original plant geometry. Preuksakarn *et al.* [21] later exploited the space colonization approach and applied to point cloud data instead of random (or user defined) set of points, thus improving the result regarding the biological relevance of the reconstructed plant.

The overall technique is a local optimization based strategy. In each iteration a local neighbourhood is considered, and no backtracking is possible to *correct* a wrong move or to refine the particle search from a global point of view. Although

the space colonization approach can produce wrong branching geometry, it can perform well if prior knowledge of the plant geometry is embedded along with the algorithm (we discuss this type of method in the later stage of the pipeline).

2.4 Step-4 : Skeleton refinement via local and global optimization

With the motivation to *improve* or *refine* the skeleton structure obtained in the previous step, different types of techniques are proposed in the literature. Refinement refers to deforming the skeleton according to the botanical rules in order to follow the geometry of the original input point cloud data. While some of the skeleton refinement techniques are optimization based, others are ad-hoc and locally heuristic in nature. The general goal of all these type of approaches is to make the skeleton biologically relevant.

Livny *et al.* [18] proposed a series of optimization techniques to improve the imperfect skeletal structure (contaminated by noise, outliers and missing data). The method is based on the initial Riemanian graph structure and the particle flow method (state transition sequence $1 \rightarrow 3 \rightarrow 4$ in Figure 1). Initially, a minimum weight spanning tree is computed from the graph by Dijkstra's shortest path algorithm. This is called the initial Branch Structure Graph (BSG). In the context of BSG, the following attributes are defined. A *simple branch* is defined as the type of branch which does not contain any junction point (where two or more branches meet). A *branch chain* is defined as the set of edges corresponding to a *simple branch*. Each node of the BSG is assigned a weight, which is equal to the size of its subtree. At this stage, the BSG hardly looks like a real tree structure due to the lack of *smoothness* in the branching structure, noise and occlusion. A number of biological constraints are proposed on the BSG structure to *refine* the branching geometry, and a series of global optimization strategies are used to apply the constraints on the BSG.

For skeleton refinement, the following criteria is imposed: the *branch chains* should be smooth, which ensures that the bending angles should be small. This is achieved by using the notion of *orientation field* [16] on the BSG vertices. Consider a BSG denoted as $\mathcal{G}$. An orientation field is a map associating each vertex $v \in \mathcal{G}$ with a vector $\overrightarrow{\mathcal{O}}_v \in \mathbb{R}^3$. This vector represents an estimated direction for moving each vertex v, so as to smooth out the skeleton. Given a vertex $v \in \mathcal{G}$ having parent v_p and the vector $\overrightarrow{v_p v}$ along the edge $e(v_p, v)$, each $\overrightarrow{\mathcal{O}}_v$ is computed to minimize the following two factors: i) the change in direction with the vector $\overrightarrow{v_p v}$, ii) the change in direction with the orientation $\overrightarrow{\mathcal{O}}_{v_p}$ of the parent node. The minimization is performed simultaneously over all the vertices of the graph $\mathcal{G}$ in a least square fashion [18]. This leads to an optimal orientation $\mathcal{O}_v^*$ at each vertex. Then the algorithm updates the vertex positions on the basis of the computed orientation $\overrightarrow{\mathcal{O}}_v$. Further refinement is obtained by performing similar type of optimization considering the center of the edges in the skeleton graph. Finally, allometric rules are used to reconstruct
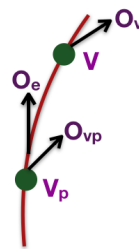
the branch thickness. Figure 2 shows some results of realistic plant modelling by this approach.

The optimization technique does not incorporate prior botanical knowledge of plants into the framework, and thus suffers from botanical inconsistency of the model. For example, the skeleton points might get shifted away from the original branch point cloud after the optimization. Also, there is no strategy for the skeleton points to follow the centerline of the branches.

Fig. 2: Modelling results from the approach of Livny *et al.* [18]. Two examples are shown in two rows. In each row, the first images shows a photo of the scene, the second image is the point cloud of a (zoomed-in) part the scene, the thrid image is the reconstructed tree, and the last one is the rendered model with the leaves.

In a similar line of work (state transition sequence $1 \rightarrow 3 \rightarrow 4$ in Figure 1), Wang *et al.* [36] proposed a combined local and global optimization technique to refine an existing skeleton, especially to handle the case of occlusion/missing data. Given an input point cloud $\mathcal{P} = \{p_i\}$ in 3D point cloud format obtained from terrestrial laser scanning (having missing/incomplete points), first a rough skeleton is extracted from the data using a standard skeletonization technique. The missing parts of the data results in disconnected components in the skeleton, which are connected together by a spanning tree based approach. However, this approach does not necessarily support the actual topology of the structure. In the next step, skeletonization of the input data is performed again by *contracting* the original point cloud to the previously obtained skeleton points using the Laplacian contraction method of Cao *et al.* [25]. The idea of Laplacian contraction method is to first *contract* or *collapse* the input point cloud to a 1D structure of

points using a Laplacian smoothing technique [26]. An energy function is defined which consists of two terms: the first term removes geometrical details along the surface normal direction, and the second term preserves the shape of the original point cloud [26]. The advantage of performing two stage skeletonization process is to exploit the strengths of both type of techniques. While the skeleton obtained in the first step roughly *completes* the missing parts of the data, the contraction technique in [25] can not handle missing data problem. On the contrary, while rough skeletonization does not guarantee to preserve the topology of the object shape, the contraction technique in [25] performs better in this aspect. Further processing is performed on the skeleton to fine tune the result by minimizing an objective function which takes into account the branch dominant direction and point density factors. In the optimization process, the skeleton point cloud $\mathcal{S} = \{s_j\}$ is pushed to move by means of a repulsive force $\mathbf{F}_r$ away from the original point cloud, while the original point cloud is forced to contract toward towards the skeleton point cloud by means of a constraint force $\mathbf{F}_s$. The optimal skeleton points are obtained by solving the following energy minimization problem,

$$\min_{\mathcal{S}} \sum_{p_i \in \mathcal{P}} ||\mathbf{F}_r(p_i, \mathcal{S}) + \lambda\mathbf{F}_s(p_i, \mathcal{S})||^2, \tag{1}$$

where λ controls the trade-off between the two terms [36].

In order to prevent large displacement of the points during the optimization process, the points are constrained to move very short distances and an iterative procedure is used to refine the displacements. Figure 3 shows some results from the technique.

In a recent work, Wu *et al.* [24] developed a skeleton refinement technique specifically for maize plants focusing the local issues like centeredness and maintaining the geometry of the original point cloud. Like the previous approach, the initial skeleton is obtained by a Laplacian based contraction method of Cao *et al.* [25]. In order to adaptively sample the contracted skeleton points, principal component analysis is performed to determine the intersection points of tree branches (junctions), which are used as keypoints. The sampling is performed in higher density near the keypoint regions. After connecting the skeleton points by edges using nearest neighbor strategy, a final refinement step is performed in order to alleviate the problem of *zigzag* structure in the skeleton (Figure 4). Assuming that the points near the junctions are mainly responsible for the *zigzag* structure, a least square straight line is fitted locally to the stem skeleton points except the points near the junction. This removes the *zigzag* locally. For the leaf vein skeleton points, first a tangent line is constructed through the adjacent points. Then a cutting plane perpendicular to the tangent line is projected on the line, and the central point of the intersection is considered as the refined skeleton point. The model is based on the geometry of maize plant and not reported to work on other types of species.

In general, the above mentioned skeletonization algorithms have their strengths and limitations based on the type of application. There is no algorithm that can
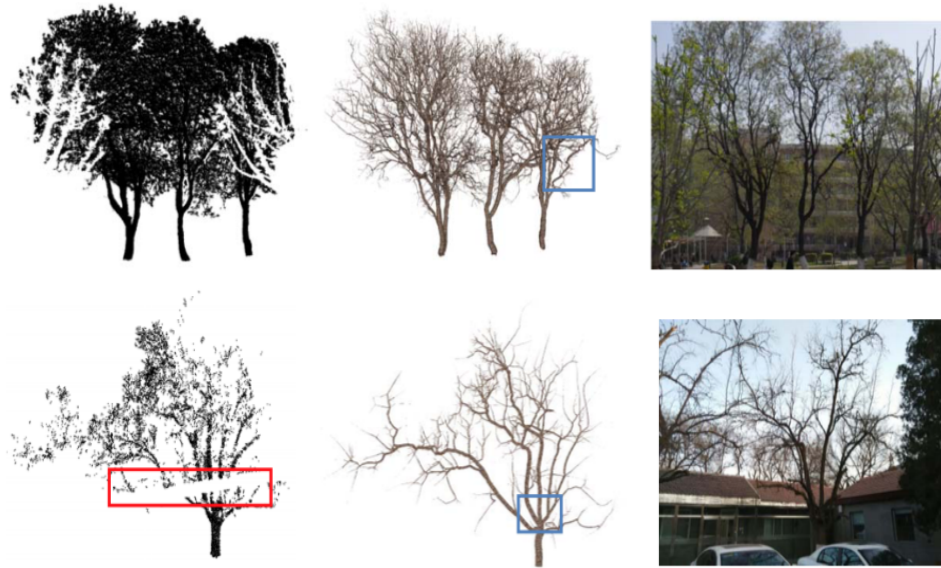
Fig. 3: Handling missing data in skeletonization by Wang *et al.* [36]. Two rows shows two examples. In each row, the left image is the raw point cloud, the middle image is the reconstructed 3D model, and the right image is the photograph of the original tree.
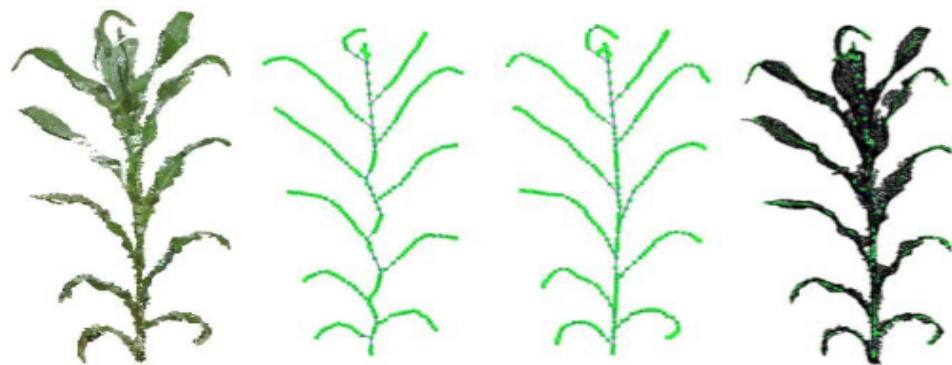

Fig. 4: Skeletonization approach of Wu *et al.* [24]. The left one is the original plant, next one is the skeleton extracted by Laplacian contraction [25] which suffers from the *zigzag* structure. The third one is the result of proposed skeletonization algorithm that solves this problem. In the right most image, the skeleton points are embedded on the original plant for visualization purposes.

handle any type of data in a robust manner. Basic skeletonization techniques discussed in the first stage of the pipeline can be used for general purpose applications, whereas the refined skeletons may suit better for specific cases. Local optimization techniques tend to focus on the issues like centerdness criteria, whereass the global optimization techniques focus on maintaining the overall shape of the skeleton. Ideally, the trade-off between two types of optimization techniques is the crucial factor in the performance of the skeletonization process.

Apart from the above mentioned skeletonization algorithms which are mostly focused on plants, some other types of general skeletonization techniques have been successful on plants as well to some extent. Some notable works in this aspect include [27] and [28]. The technique proposed in [27] is based on the notion of generalized rotational symmetry axis. The local neighbourhood of each point on the object is assumed to be cylindrical, and surface normal information is exploited to compute the skeleton. For handling the (non-cylindrical) regions where different (cylinderical) object parts join, a spatial coherence strategy is used. The L1 median skeleton technique [28] extended the idea of classical L1-median statistics by introducing a regularization term in the energy function. For a recent survey of 3D skeletonization algorithms, interested readers are encouraged to read [12].

2.5   Step-5 : Organ Surface and Volume Reconstruction

While skeletonization of plant point cloud data provides the basic geometry of the branching structure of plants, reconstruction of different organs in terms of volumetric (e.g. for branch thickness) and surface (e.g. for leaf, flower) reconstruction is the ultimate goal of plant modelling. There are different types of techniques to achieve this. In the subsequent sections, we discuss about recent approaches of volume and surface reconstruction techniques by categorizing the algorithms as follows (for most of these algorithms, the transition sequence is $1 \rightarrow 5$ in Figure 1).

Reconstruction using Real Plant Parts: The main idea of this type of technique is to dissect a plant into different parts, followed by scanning of individual parts in an offline manner, and then reconstruct the original tree geometry using these scanned parts. The technique performs skeletonization as an intermediate step, and thus the state transition sequence is $1 \rightarrow 3 \rightarrow 5$ in Figure 1.

Yin *et al.* [34] proposed a method to reconstruct plants consisting of mainly leaves. The motivation of the work is to handle the case of occlusion explicitly. Due to the cases of heavy self occlusion, inaccessible parts, diverse topologies, slim petioles and complex foliage geometry, it is extremely hard to obtain a complete model of the plant via conventional acquisition techniques (like using laser scanner to scan the whole plant from multiple views and then reconstructing the overlapping views to obtain a full 3D model). To handle this problem, a two-step approach is proposed. In the first step, the plant is scanned from several overlapping views, which are then registered to obtain a *coarse* 3D point cloud

model of the plant. In the next step, the plant is disassembled into disjoint parts. Each of these parts are scanned at a finer level of detail, and a mesh is reconstructed for each part. Then using some user assistance, each part is placed near the corresponding part in the coarse 3D model, and point set registration is performed to fit the part in the coarse model. This process is repeated for all the parts, which results in *improving* the coarse 3D model to obtain a complete 3D model of the plant.

Mesh reconstruction of the individual plant parts are performed as follows. After dissecting the k-th part from the plant, it is scanned to obtain a point cloud $\mathcal{S}_k$. The point cloud is converted into a skeleton via the L1-median skeleton method [28] and the points are resampled to the desired number of points in an uniform manner. Then for each skeleton point p_i, a slicing plane perpendicular to the skeleton curve is computed, and the original points from $\mathcal{S}_k$ are projected to the closest slicing plane. This forms a set of cross-sectional slices (s_i) along the skeleton. Now the points in this model are classified as leaf or stem (or petiole) points by a segmentation technique proposed by Li *et al.* [37], which is an extension of the classical graph cut segmentation technique [38] for 3D case. Then each slice s_i in the stem is approximated by a circular Non-Uniform Rational Basis Spline (NURBS) curve, and the slices in the leaf are approximated by closed NURBS curves (except the tip of the leaf). The thickness of the closed curves are modelled as gradually decaying values from the center to the boundary. Finally, the profile curves are joined together to form a manifold mesh $\mathcal{M}_k$ by a *sweeping reconstruction* technique [39]. The idea is shown in Figure 5.
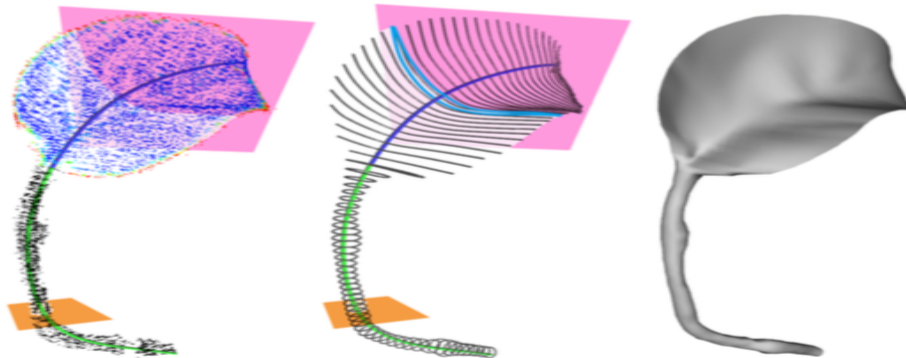

Fig. 5: Part reconstruction technique proposed in [34]. Left: raw point cloud labelled as leaf (blue) and stem (green), Middle: circular NURBS profile curve for each slice, Right: reconstructed manifold mesh using the *sweeping reconstruction* technique [39].

Once all the individual parts are reconstructed, each of these part meshes $\mathcal{M}_k$ are registered to the coarse plant model $\mathcal{P}$. A joint global and local geometry reconstruction method is used to perform the registration task. The global step of the registration starts with an interactive procedure. The user places a part close to its actual location in the coarse model. Then a point-by-point correspondence set is obtained by nearest neighbour strategy. Using this coarse correspondence set, the part mesh is transformed (rotation and translation) towards the full model. The optimal set of transformation parameters are obtained as,

$$\underset{r_i,t_i}{\arg\min} \left(D(\mathcal{M}_k(r_i,t_i),\mathcal{P}) + \lambda L(\mathcal{S}_k(t_i)) \right), \tag{2}$$

where the first term $D(\cdot,\cdot)$ is the data fitting term which transforms $\mathcal{M}_k$ by the transformation parameters (r_i,t_i), the second term $L(\cdot)$ is the regularization term which transslates the skeleton curve by t_i, and λ is the weighting factor of the regularization (or smoothness) term. The data term computes the weighted sum of the inner correspondence distances, and the smoothness term ensures the uniform point distribution along the curve skeleton with the skeleton length preservation.

In the local registration step, similar type of objective function is created. The objective functions are minimized by a standard framework (Broyden-Fletcher-Goldfarb-Shanno solver). Figure 6 shows the whole pipeline.

(a) Photo.   (b) Incomplete scan.   (c) Leaf templates and branches reconstructed.

(d) Initial placement.   (e) Global skeleton-driven reg.   (f) Local skin-guided align.   (g) Final reconstruction.
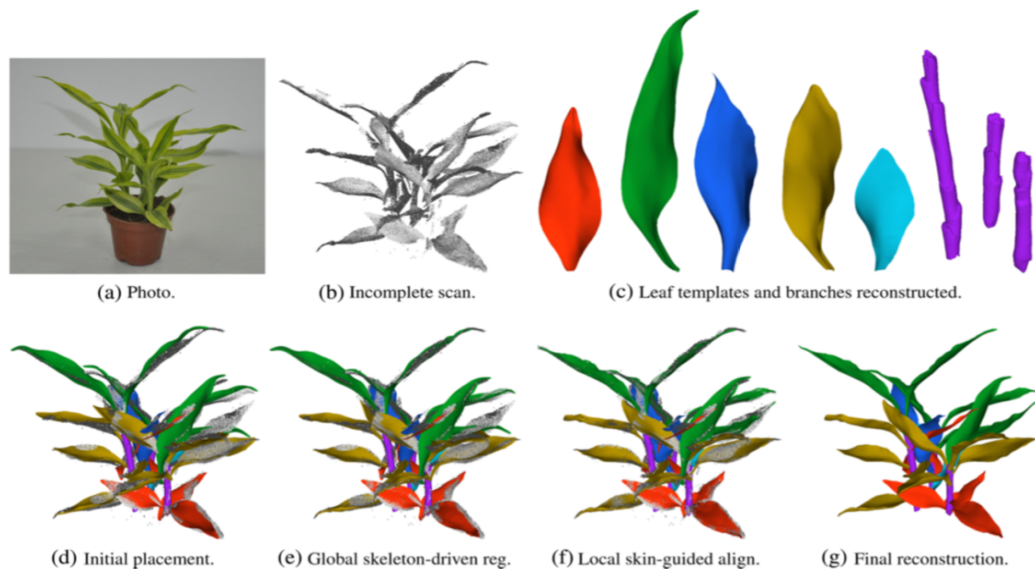
Fig. 6: The pipeline of the method proposed in [34].

In a similar line of work, Xie *et al.* [33] proposed a semi-automatic modelling approach using scanned tree parts. The method is designed for computer graphics applications, where the user can choose certain parts of the tree from a number of parts stored in the database, and then the algorithm automatically constructs the whole plant model by *joining* these parts in a biologically meaningful manner (state transition sequence $1 \rightarrow 5$ in Figure 1).

Initially, 15 different tree species are cut into about 200 tree parts. Each of these parts are scanned separately to obtain point cloud of each tree part, which are stored in a database. Next, the user chooses to reconstruct a particular tree from a template photograph, and places the required tree parts from the database in the 3D space according to the photograph. This represents the basic structure of the tree, which remains in a disconnected or incomplete stage (Figure 7).

Fig. 7: The modelling process proposed by Xie *et al.* [33]. Initially, *tree cuts* are placed in space in an intuitive manner, which will be joined in the next phase to complete the tree structure. Two examples are shown. In each example, the first image is the photograph of the real tree and the second image is the initial stage of placing the *tree cuts* in space.

In the next phase, the *tree cuts* are connected to form a realistic tree in biologically relevant manner. Given a tree cut, first the Euclidean distances from the end point of the current tree cut to the end points of all the cuts are computed. The tree cut having the minimum distance is considered as the nearest object from the current tree cut. Now the transformation parameters (rotation and translation) between the contours (a contour is assumed to be closed, and thus forms a *"loop"*) of the two tree cuts are estimated to obtain point to point correspondence between the two parts. In order to estimate the branch surface between the two parts, an interpolation strategy is used. A Hermite curve is computed between the center points of the two contours, and a number of *loops* are generated along the curve by keeping the loop centers to be perpendicular to the curve direction. Incorporation of additional bifurcations (apart from the bifurcation data in the database) are introduced in the model by means of some user assistance. The user gives input for the position of a bifurcation, and the algorithm automatically connects different tree parts associated with the

bifurcation in a automatic manner by an interpolation strategy. The branching diameters and angles are estimated using standard *allometry rules* to add realism in the reconstruction process. Figure 8 shows some examples of reconstructed trees from the tree cuts.
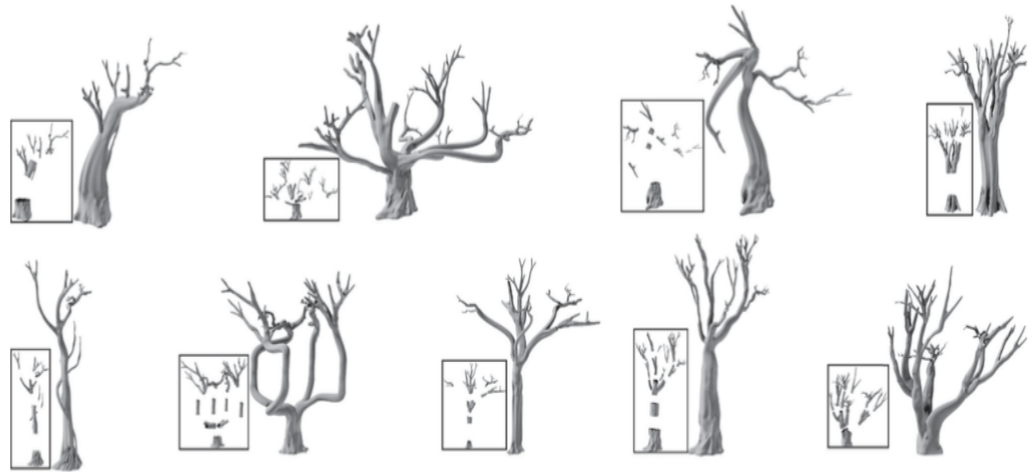

Fig. 8: Results of connecting the tree cuts to reconstruct full 3D plant model [33].

One drawback of the above approach is that, the method demands a number of tree parts to build the database. Also, it needs some user assistance in order to work properly. Although modelling the missing branch parts are approximated by parametric curves, it is not fully supported by biological relevance of the structure. The technique is not fully data driven in the sense that the input point cloud data of the whole plant is not used, and thus the algorithm does not focus on reconstructing the original plant geometry.

Hybrid Approach The hybrid approach is based on a combined framework of procedural (i.e. the type of modelling that is based on a set of predefined rules to produce the output) and data driven modelling techniques (state transition sequence $1 \rightarrow 5$ in Figure 1). The *Inverse Procedural Model* approach [29] is a hybrid approach where the main idea is to estimate the parameter of a procedural model that best represents the input scan data, instead of defining rules that generate a tree model as typically done in classical procedural modelling techniques. Each set of parameter defines the general architecture of a particular species, and by varying the parameters, different varieties of the species in different environmental conditions can be modelled. The model is based on a set of 24 parameters which describe the geometry of a tree. These parame-

ters include branching angle, internode distance, number of lateral buds, etc. which are based on geometric and environmental factors acting on the plant. Several biological assumptions/constraints are employed in the modelling process to mathematically represent different entities of the tree. For example, the apical bud is assumed to be located at the tip of the plant shoot. The orientation of the apical bud with respect to the shoot is modelled as the following polar and azimuthal angle in a spherical coordinate system,

$$
\begin{aligned}
\theta &\sim \mathcal{N}(0, \sigma_v), \\
\phi &\sim \mathcal{U}(0, 2\pi),
\end{aligned}
\tag{3}
$$

where σ_v is the apical angle variance parameter, $\mathcal{N}$ and $\mathcal{U}$ are normal and uniform distributions respectively. So by varying σ_v, it is possible to generate different branching structures as shown in Figure 9. In this way, the parametric model of the tree is defined by using the selected set of parameters. In order to find the parameters that maximize the similarity measure between the input data and the model, a number of trees are generated by perturbing the discrete values of the parameters empirically by the user. Then, each of these tree is compared with the input tree by a similarity measure based on three types of distances: *shape distance*, *geometric distance*, and *structural distance* (refer to Sec. 3 for details). However, there can be infinite number of tree models that can be generated by different combinations of the parameters. The best set of parameters corresponding to the best matching tree is obtained using a Monte Carlo technique.

Fig. 9: The effect of apical angle variance parameter σ_v in the model proposed in [29]. The angles are increased from left to right image.

Recently, Guo *et al.* [35] proposed a hybrid approach to the reconstruct plant geometry with the motivation to alleviate the need of tuning of large number of

parametrs in the previous approach. The technique extends the space colonization technique of Preuksakarn *et al.* [21] by introducing an L-system type rule based framework in the model.

Initially, a technique is proposed to generate high quality 3D point cloud data from a sequence of overlapping images around a plant. Next, a parametric model is fitted to the point cloud by using a rule-based method. The model is based on 5 parameters to represent the skeletal structure of the tree: internode length (l), roll angle (ϕ), branching angle (ψ), growth units (ρ), and diameter coefficient (ρ). Starting from the root/bottom of the plant, the model simulates a set of rules by certain assumptions. Basically the rules make use of the points in a biologically meaningful manner. From the seed locations, the following set of rules are used,

$$
\begin{aligned}
\text{Seed}(\mathbf{p}, \mathbf{v}) &\rightarrow A(\mathbf{p}, \mathbf{v}), \\
A(\mathbf{p}, \mathbf{v}) &\rightarrow \{\text{Metamer}(\mathbf{p}', \mathbf{v}')*\} \, A(\mathbf{p}'', \mathbf{v}''), \\
\text{Metamer}(\mathbf{p}, \mathbf{v}) &\rightarrow \text{Internode}(\mathbf{p}, \mathbf{v}) \, L(\mathbf{p}', \mathbf{v}'), \\
L(\mathbf{p}, \mathbf{v}) &\rightarrow A(\mathbf{p}', \mathbf{v}'),
\end{aligned}
\tag{4}
$$

where $\mathbf{p}$ is the seed position and $\mathbf{v}$ is the orientation, which indicates the growth direction of each bud. Initially the growth direction is upward, and is adjusted according to the branching structure and roll angles at each stage. $A(\mathbf{p}, \mathbf{v})$ is the *kernel* of the growth process, which results in a chain of metamers (or a growth unit) represented by $\{*\}$ in the rules. At any particular stage, the position of a metamer is computed from the previous metamer, orientation and length. The orientation is also updated by the phyllotaxy (roll) angle. In the implementation level, the data points are selected similar to the technique of space colonization algorithm [20, 21]. Finally, standard allometric rules are applied to compute branch diameter and length.

Other Types of Approaches: Zhang *et al.* [32] proposed a technique by first reconstructing the visible parts of the scanned input point cloud data, and then synthesized the non-visible parts by a shape prediction model. The approach is a combination of geometry reconstruction at different hierarchical levels (from fine to coarse), and attempts to model the multiscale aspect of tree reconstruction.

Initially, all the points in the input point cloud data are labelled as *unlabelled*. Then starting from an user specified point, an initial cylinder is fitted to the local neighbourhood. Once a set of points are fitted within a cylinder, these points are labelled as *processed*, and are not considered any more for processing. An iterative process is continued to fit successive cylinders until no cylinder can be fitted any more to the data (*cylinder marching* process). This process is based on the assumption that branch shapes are cylindrical with gradually varying radii. If the number of connected cylinders go beyond a threshold value, the cylinders constitute a single branch of the tree. At this stage, visible branches of the tree are obtained from the fitted cylinders. To reconstruct the non-visible branches of the tree (due to the occlusion), the following heuristics is used. At

the end of cylinder marching process, remaining unlabelled scattered points are classified as tree crown points, which mostly belong to the leaves at the tip of the branches. Then the classical particle flow technique is used to model the non-visible branches by considering the crown points as the source and nearest main branch point as the destination. The direction of flow of the particles are computed using the conical search operation of the space colonization algorithm as discussed earlier. Finally, some standard techniques are applied to produce texture effect, mesh model, and leaf configurations.

One drawback of the technique is that, lot of heuristics are used in the whole process. Also, tuning of parameters are very important to obtain the desired result.

Flower Geometry Reconstruction While the goal of volume reconstruction is to model mainly the branching system the the plant, reconstruction of organs like flower and leaf require surface based techniques. Because of the complicated thin structure of the petals and their tight configurations along with high level of occlusions, geometry reconstruction of flowers is an important as well as a hard problem. Unlike the case of branching structure reconstruction, there are only a handful of techniques which take into account the real data to model the flower geometry.

Recently, Yan *et al.* [40] proposed a model to reconstruct the flower geometry from single photograph. The model is based on the assumption that the 3D shapes of different petals are roughly the same. Initially, modelling flowers having single layer of petals is considered and then the idea is extended for flowers having multiple layers of petals. First, the petals of a single layer flower are segmented in an interactive way by the well known *GrabCut* [41] technique. Individual petals are located from the segmented image using the following technique. First, the center of the flower is located by user assistance ((red dot in the first image of Figure 10)). The the contour of the segmendted flower is traversed and distance from each contour point to the center is computed. *Valleys* in the sequence of the distance values indicate the intersection points between adjacent petals (yellow dots in the middle image of Figure 10). The contour of each petal is identified by simulating the classical particle flow technique, where a particle is traced from the intersection point towards the flower center by following the edge of the petal (similar to the *Livewire* technique discussed earlier). The tip of each petal is located at the midpoint of the contour (green dots in the right image of Figure 10).

In the next phase, the underlying surface of the petals are modelled by a 3D cone, assuming the flower structure to be of conical shape. The 3D cone is fitted to the extracted petals, where the apex of the cone is positioned at the center and the base is aligned with the petal tips. This cone is used as a special case of a *surface of revolution*, which can be defined by rotating a curve around the main axis of the flower. By an iterative procedure, each petal contour is fitted to the surface of revolution framework, and the template of individual petal is obtained. Once the petal contour is obtained, a 3D mesh is constructed by

Fig. 10: Petal contour detection technique proposed in [40]. (Left image) Initially, the center of the flower is located by user assistance (red dot). (Middle image) The intersection points of the petals are located as yellow dots. (Right image) The petals tips are located as the green dots.

sampling points enclosed in the contour and triangulating these points. However, at this stage the mesh does not reflect the geometry of bending of each petal. In order to facilitate this feature, the mesh needs to be deformed to approximate the petal shape. The deformation *drags* the vertices on the boundary of the mesh, so that the differences between the observed contours and the projections of the corresponding mesh is minimized. This is performed by minimizing the following energy,

$$\mathcal{E} = \lambda_{\mathrm{geo}}\mathcal{E}_{\mathrm{geo}} + \lambda_{\mathrm{con}}\mathcal{E}_{\mathrm{con}}, \tag{5}$$

where $\mathcal{E}_{\mathrm{geo}}$ is the geometric preserving energy that maintains the geometric features of the mesh, $\mathcal{E}_{\mathrm{con}}$ is the contour fitting energy that drags boundary vertices to their new locations, and $\lambda_{\mathrm{geo}}, \lambda_{\mathrm{con}}$ are the weighting factors for the two terms. The deformation is performed in an iterative manner. Figure 11 shows some results of the modelling approach.

Fig. 11: Reconstruction of flower model from single image [40]. Two models are shown, where the left image is the original flower, the middle shows the reconstructed mesh, and the right is the rendered result.

In a similar type of work, Zheng *et al.* [42] presented an algorithm to dynamically track a blooming flower over time. The idea is based on the reconstruction of flower geometry and deformation of the model over time via a template model.

During the blooming stage, the flower is scanned over the time span $1 \cdots T$ to obtain a sequence of point clouds $\mathbb{Q} = \mathbb{Q}_{1:T}$. During the early stage, interior petals are completely occluded, whereas during the later stage, exterior petals decay, bend and twist heavily. So, none of the scans can actually represent the whole flower geometry by itself. Initially, one intermediate frame is selected manually which is considered as the *key frame*, and the geometry $\mathbb{M}$ of the full flower is reconstructed from this frame by in an interactive manner (assuming that the petals are at least partially visible at the selected key frame). The main idea of the technique is to use $\mathbb{M}$ as the template, and fit this to the point cloud in the adjacent frames by deformation. The *track and fit* operation is formulated as Expectation Maximization (EM) framework, where the E-step computes the correspondence between the template and the captured point cloud, and the M-step updates the vertex locations based on the computed correspondence of the vertices. The tracking is performed both forward and backward in time, which yields a series of 3D model sequence $\mathbb{M}_{1:T}$, which represents the flower blooming process.

Let $\mathbb{Q}_t$ represents the captured data at time t and $\mathbb{M}_t$ represents the deformed template mesh model (which we are seeking) of the original template model $\mathbb{M}$. In the EM framework, $\mathbb{Q}_t$ is assumed to be the observation of a Gaussian Mixture Model (GMM), whose centroids are the vertices of the unknown mesh $\mathbb{M}_t$. The problem is formulated as Maximum a Posteriori (MAP) problem to obtain the deformed mesh vertices $\mathbb{M}_t^*$, which is obtained by maximizing the probability of the observation,

$$\mathbb{M}_t^* = \arg\max_{\mathbb{M}_t} p(\mathbb{Q}_t|\mathbb{M}_t)p(\mathbb{M}_t), \tag{6}$$

where $p(\mathbb{Q}_t|\mathbb{M}_t)$ is the likelihood term, and $p(\mathbb{M}_t)$ is the prior term. During the E-step, vertex correspondences between $\mathbb{M}$ and $\mathbb{Q}$ are computed. Instead of computing the correspondence using the point cloud as a whole, these are classified into different parts (petals) where each part $\mathbb{Q}^k \in \mathbb{Q}$ corresponds to a part $\mathbb{M}^k \in \mathbb{M}$. The classification is performed by standard Gaussian Mixture Model (GMM) clustering strategy. Assuming that a point $q_j \in \mathbb{Q}$ is normally distributed around $m_i \in \mathbb{M}$ as $q_j \sim \mathcal{N}(m_i, \sigma_i)$ with covariance σ_i, the probability of q_j given m_i is,

$$p(q_j|m_i) = \frac{1}{\sqrt{(2\pi)^3|\sigma_i|}} \exp\left(-\frac{1}{2}(q_j - m_i)^T \sigma_i^{-1}(q_j - m_i)\right). \tag{7}$$

The probability that a point q_j belongs to $\mathbb{Q}^k$ is computed as,

$$p(q_j \in \mathbb{Q}^k) = \frac{\sum_{m_i \in \mathbb{M}^k} \phi_i p(q_j|m_i)\nu_i}{\sum_{m_i \in \mathbb{M}} \phi_i p(q_j|m_i)\nu_i}, \tag{8}$$

where $\phi_i = \sum_{q_j \in \mathbb{Q}^k} p(q_j|m_i)$, and $\nu_i \in \{0, 1\}$ is called the *visibility term*, which is set to 1 if the point is visible (within a threshold distance from the mesh), otherwise it is set to 0.

Now, for each part (petal) k, the correspondence between $\mathbb{Q}^k$ and $\mathbb{M}^k$ is computed as a point correspondence matrix $\mathbb{Z} : \mathbb{M}^k \to \mathbb{Q}^k$, where each element $\mathbb{Z}_{ij}$ of $\mathbb{Z}$ is computed as,

$$\mathbb{Z}_{ij} = \frac{\phi_i p(q_j|m_i) \nu_i}{\sum_{m_i \in \mathbb{M}^k} \phi_i p(q_j|m_i) \nu_i}. \tag{9}$$

Once the correspondences of vertex locations are estimated for each petal, now the task is to optimize the vertex locations in $\mathbb{M}$, so that $\mathbb{M}$ fits better to the data $\mathbb{Q}$ (the M-step). This is formulated as the following stochastic minimization problem,

$$\arg \min_{\mathbb{M}} (-\log p(\mathbb{M}|\mathbb{Q}, \mathbb{Z}) - \log p(\mathbb{M})). \tag{10}$$

The first term in the above equation is the data term, where the goal is to minimize the distances between mesh vertices and their corresponding points. Along with an additional penalty to ensure that the mesh follows the contour of the petal, the data term is written as,

$$-\log p(\mathbb{M}|\mathbb{Q}, \mathbb{Z}) = \sum_k (\omega_1 D(\mathbb{Q}^k, \mathbb{M}^k) + \omega_2 D(\mathbb{Q}_b^k, \mathbb{M}_b^k)), \tag{11}$$

where $\mathbb{Q}_b^k$ and $\mathbb{M}_b^k$ are the boundary points, which are not taken into consideration at the initial stage of the blooming process since the petals are not sufficiently separated at this stage. ω_1, ω_2 are the weighting factors for the two terms. The second term of Equation 10 is the prior term which regularizes the solution. This term is modelled as a combination of 3 different types of priors as follows,

$$-\log p(\mathbb{M}) = \mathcal{E}_{\text{shape}} + \mathcal{E}_{\text{collision}} + \mathcal{E}_{\text{root}}. \tag{12}$$

The first term is used to preserve the flower shape. Using some user intervention, a reliable template mesh is created and the deformation of $\mathbb{M}^k$ is constrained by this template mesh. In the second term, avoidance of the penetration of two petals are ensured. Whenever a penetration is detected, the point is moved to another location so that the surface normal directions are not violated. The third term ensures that the root (or the base) point of each petal does not change during deformation. Finally the energy is minimized by alternate optimization strategy using different types of nonlinear least square solvers. Figure 12 shows some results of the modelling.

One of the drawback of the method lies in it's limitation to model fine geometric details. Also, large deformation of petals will result in poor registration of the point cloud, which will affect the overall accuracy of the reconstructed model.

Fig. 12: Example of reconstruction of a blooming Golden Lily. The top row shows the point cloud and the bottom row shows the rendered reconstructed model [42].

3   Evaluation Techniques

So far we have discussed different types of modelling techniques which aim at reconstructing the geometry of plants from real data. However, an important aspect of reconstruction is to measure the quality of the solution. From computer graphics perspective, the results are analyzed mostly on qualitative or visual assessements. However in biological applications, quantitative analysis of the results is extremely important. Given a reference plant structure (the groundtruth), different types of measures are proposed in the literature to quantify the similarity between the reconstructed model and the reference model. A naive approach is to measure quantities like crown volume, total branch length, etc. and then compare the two models by the difference of these quantities. However, this type of method cannot truly asses the quality of reconstruction of tree topology. Ferraro *et al.* [30] proposed an advanced metric to compare 2D tree structures by means of *edit distance*. The distance is computed as the minimum cost of a sequence of edit operations that transforms an initial tree into a target tree. Boudon *et al.* [43] adopted the idea of 2D edit distance into the 3D case to compare two trees. An optimization framework is proposed to compute geometric and structural similarities of two trees. Initially, the scales (or the resolutions) of the scan data of two trees are homogenized. Then Similar elements of the trees are determined by adopting the Hausdorff distance between their skeleton curves. Structural similarity between the two trees is computed by computing the similarity of the organization of the edges.

Stava *et al.* [29] proposed a similarity measure by introducing two different types of distance measure along with the edit distance [30]. *Shape distance* is used to measure the difference between the overall shapes of the trees. Different types of descriptors (e.g. height, radius, etc.) are evaluated in the shape distance function as,

$$\delta = 1 - \exp\left[1 - \frac{(\xi_{\tau_i} - \xi_{\tau_j})^2}{2\sigma^2}\right], \tag{13}$$

where ξ_{τ_i} and ξ_{τ_j} are the descriptors of the two trees and σ is a normalization factor which is set empirically. Another type of measure, the *geometric distance* computes the difference between global branching structure of the two trees from some sample points over the trees. Several parameters like branch angle, branch slope with respect to the horizontal plane, etc. are computed for all the sample points over the tree, and the descriptor is defined as the weighted mean and variance of these samples. The geometric distance is computed using these descriptors. The final similarity measure $\bar{\delta}(\tau_i, \tau_j)$ of two trees τ_i and τ_j is computed as the sum of the three measures, along with the weighting factors that control the influence of each measure.

On a different type of evaluation strategy, Guo *et al.* [35] used two types of evaluation metrics: *model based* and *scan based*. In the model based strategy, synthetic plant models are used as ground-truth and a number of views are generated from the model. Then the similarity distance as proposed in [29] are used as the evaluation metric. In the scan based strategy, the comparison is made directly on the real data. For every point in the reconstructed model, the distance to the nearest point in the actual scan data are computed. The average and standard deviation of the distances are used as performance measurement. If the values are lower, then the performance is treated as better, whereas higher values indicate worse performance.

4   Conclusion

In this chapter, we have broadly concentrated on the topic of geometry reconstruction of plants from real world data. A number of state-of-the-art techniques are discussed, along with their strengths and limitations. We infer from the thorough literature survey that there handful of techniques which actually take care into account the actual biological relevance of the results. Also, not much work has focused on exploiting the prior botanical knowledge in the modelling framework. Given the huge varieties of plant species, there is lot of scope of geometry modelling to accurately reconstruct the plant geometry in a robust manner. Ideally, the ultimate goal of geometry reconstruction will be to develop a generalized algorithm which will be able to handle a diverse varieties of species that can work in real time.

Acknowledgement

References

1. P. Prusinkiewicz and A. Lindenmayer, "The Algorithmic Beauty of Plants", Springer-Verlag, 1990.
2. R. Karwowski and P. Prusinkiewicz, "Design and Implementation of the L+C Modeling Language", Electronic Notes in Theoretical Computer Science, Vol. 86, pp. 134-152, 2003.
3. P. Prusinkiewicz, L. Mündermann, R. Karwowski and B. Lane, "The Use of Positional Information in the Modeling of Plants", In Proceedings of SIGGRAPH, 2001.
4. C. Godin, E. Costes and H. Sinoquet, "A Method for Describing Plant Architecture which Integrates Topology and Geometry", Annals of Botany, 84: 343-357, 1999.
5. I. Shlyakhter, M. Rozenoer, J. Dorsey, S. Teller, "Reconstructing 3D Tree Models from Instrumented Photographs", IEEE Computer Graphics and Applications, 21(3), 53-61, 2001.
6. A. Reche, I. Martin and G. Drettakis, "Volumetric Reconstruction and Interactive Rendering of Trees from Photographs", ACM Transactions on Graphics, Vol. 23, No. 3, 2004.
7. C. Godin and H. Sinoquet, "Functional-structural plant modelling", New phytologist, 166(3), 705-708, 2005.
8. N. Fahlgren, M. A. Gehan and I. Baxter, "Lights, camera, action: high-throughput plant phenotyping is ready for a close-up", Current Opinion in Plant Biology, Vol. 24, pp. 93-99, 2015.
9. 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, 2018.
10. L. Yi, H. Li, J. Guo, O. Deussen and X. Zhang, "Tree Growth Modelling Constrained by Growth Equations", Computer Graphics Forum, 2017.
11. N. D. Cornea, D. Silver and P. Min, "Curve-Skeleton Properties, Applications, and Algorithms", IEEE Transactions on Visualization and Computer Graphics, Vol. 13, No. 3, 2007.
12. A. Tagliasacchi, T. Delame, M. Spagnuolo, N. Amenta and A. Telea, "3D Skeletons: A State-of-the-Art Report", Computer Graphics Forum, Volume 35 (2016), No. 2, 2016.
13. H. Xu, N. Gossett and B. Chen, "Knowledge and heuristic based modeling of laser scanned trees", ACM Transactions on Graphics, Vol. 26, No. 4, 2007.
14. D. Yan, J, Wintz, B. Mourrain, W. Wang, F. Boudon and C. Godin, "Efficient and robust reconstruction of botanical branching structure from laser scanned points", In Proceedings of IEEE International Conference on Computer Aided Design and Computer Graphics (CAD), 2009.
15. Y. Rodkaew, P. Chongstitvatana, S. Siripant and C. Lursinsap, "Particle Systems for Plant Modeling", In Proceedings of Plant Growth Modeling and Applications, pp. 210-217, 2003.
16. B. Neubert, T. Franken and O. Deussen, "Approximate Image-Based Tree-Modeling using Particle Flows", ACM Transactions on Graphics, Vol. 26, No. 3, 2007.
17. E. N. Mortensen and W. A. Barrett, "Intelligent Scissors for Image Composition", In Proceedings of SIGGRAPH 1995.

18. Y. Livny, Y. Feilong, M. Olson, B. Chen, H. Zhang and J. El-Sana, "Automatic Reconstruction of Tree Skeletal Structures from Point Clouds", ACM Transactions on Graphics, 29(6), pp. 151:1–151:8, 2010.

19. A. Runions, B. Lane and P. Prusinkiewicz, "Modeling Trees with a Space Colonization Algorithm", Eurographics Workshop on Natural Phenomena, pp. 63-70, 2007.

20. W. Pałubicki, K. Horel, S. Longay, A. Runions, B. Lane, R. Mech and P. Prusinkiewicz, "Self-organizing tree models for image synthesis", ACM Transactions on Graphics, Vol. 28, No. 58, 2009.

21. C. Preuksakarn, F. Boudon, P. Ferraro, J. B. Durand, E. Nikinmaa and C. Godin, "Reconstructing Plant Architecture from 3D Laser scanner data", In Proceedings of 6th International Workshop on Functional Structural Plant Models, 2010.

22. A. Bucksch, R.C. Lindenbergh and M. Menenti, "SkelTre - fast skeletonisation for imperfect point cloud data of botanic trees", Eurographics Workshop on 3D Object Retrieval, 2009.

23. V. Pascucci, G. Scorzelli, P. Bremer and A. Mascarenhas, "Robust on-line computation of Reeb graphs: simplicity and speed", ACM Transactions on Graphics, Vol. 26, No. 58, 2007.

24. 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, 2019.

25. J. Cao, A. Tagliasacchi, M. Olson, H. Zhang and Z. Su, "Point Cloud Skeletons via Laplacian-Based Contraction", in Proceedings of Shape Modeling International (SMI) Conference, 2010.

26. O. K. Au, C. Tai, H. Chu, D. Cohen-Or and T. Lee, "Skeleton extraction by mesh contraction", ACM Transactions on Graphics, Vol. 27, No. 44, 2008.

27. A. Tagliasacchi, H. Zhang and D. Cohen-Or, "Curve Skeleton Extraction from Incomplete Point Cloud", ACM Transactions on Graphics, 28(3), 71:1–71:9, 2009.

28. 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, 32(4), 65:1–65:8, 2013.

29. O. Stava, S. Pirk, J. Kratt, B. Chen, R. Mech, O. Deussen, B. Benes, "Inverse Procedural Modeling of Trees", Computer Graphics Forum, Vol. 33, Issue 6, pp. 118-131, 2014.

30. P. Ferraro and C. Godin, "A Distance Measure Between Plant Architectures", Annals of Forest Science, 57, 5/6, 445-461, 2000.

31. 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, No. 76, 2012.

32. X. Zhang, H. Li, M. Dai, W. Ma and L. Quan, "Data-Driven Synthetic Modeling of Trees", IEEE Transactions on Visualization and Computer Graphics, Vol. 20, No. 9, 2014.

33. K. Xie, F. Yan, A. Sharf, O. Deussen, H. Huang and B. Chen, "Tree Modeling with Real Tree Parts Examples", IEEE Transactions on Visualization and Computer Graphics, Vol. 22, No. 12, 2016.

34. K. Yin, H. Huang, P. Long, A. Gaissinski, M. Gong and A. Sharf, "Full 3D Plant Reconstruction via Intrusive Acquisition", Computer Graphics Forum, Volume 35, No. 1, pp. 272-284, 2016.

35. J. Guo, S. Xu, D. Yan, Z. Cheng, M. Jaeger and X. Zhang, "Realistic Procedural Plant Modeling from Multiple View Images", IEEE Transactions on Visualization and Computer Graphics, 2018.

36. Z. Wang, L. Zhang, T. Fang, X. Tong, P. T. Mathiopoulos, L. Zhang and J. Mei, "A Local Structure and Direction-Aware Optimization Approach for Three-Dimensional Tree Modeling", IEEE Transactions on Geoscience and Remote Sensing, Vol.54, No. 8, 2016.
37. Y. Li, X. Fan, N. J. Mitra, D. Chamovitz, D. Cohen-Or and B. Chen, "Analyzing growing plants from 4D point cloud data", ACM Transactions on Graphics, Vol. 32, No. 157, 2013.
38. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts", IEEE Transactions on Pattern Analysis and Machine Intelligence, 23(11), pp. 1222-1239, 2001.
39. K. Yin, H. Huang, H. Zhang, M. Gong, D. Cohen-Or and B. Chen, "Morfit: Interactive Surface Reconstruction from Incomplete Point Clouds with Curve Driven Topology and Geometry Control", ACM Transactions on Graphics, Vol. 33, No. 202, 2014.
40. F. Yan, M. Gong, D. Cohen-Or, O. Deussen and B. Chen, "Flower Reconstruction from a Single Photo", Computer Graphics Forum, Vol. 33, No. 2, 2014.
41. C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts", ACM Transactions on Graphics, Vol. 23, pp. 309-314, 2004.
42. Q. Zheng, X. Fan, M. Gong, A. Sharf, O. Deussen and H. Huang, "4D Reconstruction of Blooming Flowers", Computer Graphics Forum, Vol. 36, pp. 405-417, 2017.
43. F. Boudon, C. Preuksakarn, P. Ferraro, J. Diener, P. Nacry, E. Nikinmaa and C. Godin, "Quantitative Assessment of Automatic Reconstructions of Branching Systems Obtained from Laser Scanning", Annals of Botany, Vol. 114, No. 4, pp. 853-862, 2014.