

Full 3D Plant Reconstruction via Intrusive Acquisition

Kangxue Yin¹ Hui Huang^{1†} Pinxin Long¹ Alexei Gaissinski² Minglun Gong³ Andrei Sharf⁴

¹ Shenzhen VisuCA Key Lab / SIAT ² Tel Aviv University ³ Memorial University of Newfoundland ⁴ Ben Gurion University

Abstract

Digital capturing vegetation using off-the-shelf scanners is a challenging problem. Plants typically exhibit large self-occlusions and thin structures which cannot be properly scanned. Furthermore, plants are essentially dynamic, deforming over the time, which yield additional difficulties in the scanning process. In this paper we present a novel technique for acquiring and modeling of plants and foliage. At the core of our method is an intrusive acquisition approach, which disassembles the plant into disjoint parts that can be accurately scanned and reconstructed offline. We use the reconstructed part meshes as 3D proxies for the reconstruction of the complete plant and devise a global-to-local non-rigid registration technique that preserves specific plant characteristics. Our method is tested on plants of various styles, appearances, and characteristics. Results show successful reconstructions with high accuracy with respect to the acquired data.

1. Introduction

3D plant modeling is an important problem with a wide range of applications. In computer graphics, 3D plant models play an important role in the design of realistic game scenes, physical effects simulations, and rendering. Within the domain of plant biology, 3D plants are modeled and explored in the context of plant growth and behavior under various simulations. In agriculture, we observe a growing interest in the accurate acquisition and modeling of 3D plant models for observations and measurements in the research of pest management, fertilization, etc. Nevertheless, accurate and efficient acquisition and modeling of real world plants remain an open problem.

Various attempts have been made to create and model 3D plants from different sources. Approaches such as procedural modeling [PL96, PHL*09], image-based modeling [QTZ*06, YGCO*14], and sketch-based modeling [IOOI05, LRBP12] have been successfully applied to create realistic 3D tree and plant models. Common to these works is the creation of realistic-looking 3D plant models using some guidance and constraints from the real world. However, acquisition and modeling of the exact geometry and topology of the plant are still very difficult.

We introduce a novel framework for acquisition and model-



Figure 1: Reconstruct complete 3D plant models (center) with heavily self-occluded leaves (*Aglaonema crispum* on top) and highly-curved, non-developable leaves (*Dracaena sanderiana* at bottom). The texture mapped rendering results (right) closely resemble the plant photos (left).

† Corresponding author: Hui Huang (hhzhiyan@gmail.com)

ing the full plant geometry and topology from 3D scans. In contrast to the recently proposed X-ray tomography (CT) acquisition system [IYYI14], our process utilizes off-the-shelf 3D scanners due to their growing popularity, availability and ease-of-use. Our focus is on a modeling approach that reconstructs the full 3D plant geometry and topology from scans in an accurate and robust manner; see Figure 1.

Scanning plants is a challenging task. Plants typically consist of multi-scale structures, complex topology and delicate features. Additionally, plants present a large amount of self-occlusions as entangled leaves, hidden stems and interior parts. Hence, it is practically an impossible task to sample the full plant geometry using standard scanning techniques.

We develop a novel intrusive acquisition system which allows to reconstruct the full geometry and topology of the plant. In the first step we loosely scan the plant to form a base reference geometry. This base geometry is incomplete with large missing parts due to self-occlusions and inaccessible parts. In the next step we disassemble the plant by cutting it into disjoint parts, which we scan independently and reconstruct their exact geometry. Finally, we register all disjoint part models back into the base geometry.

The disassembly process introduces deformations in the parts, as they are removed from their original plant configuration. We assemble the parts back into the base geometry using a non-rigid registration method, which operates in a greedy manner. That is, in each iteration we register only one part to the base geometry. We consider plant characteristics and prevent self-collisions, which may occur due to the large entanglement of leaves and stems.

The overall scanning process is destructive as the plant is cut into parts. However, this allows instant access for modeling hidden geometries and topologies. In addition, dissected parts can be scanned and modeled with high accuracy, allowing the final accurate reconstruction of the complete 3D plant, which otherwise could not be made even manually due to the complexity and inaccessibility of the plant.

Our work makes two main contributions:

- we introduce a new plant acquisition pipeline, which is intrusive and acquires the full geometry of the foliage;
- we develop a robust global-to-local non-rigid registration algorithm, which uses the knowledge of the plant geometry combined with a general deformation approach to fully reconstruct the complete plant.

2. Related work

Rule-based plant modeling. 3D modeling methods for plants and trees have been an active research problem in computer graphics for more than two decades. Early works on plant modeling involved parameterized algorithms and rule-based procedural methods [PJM94, PL96]. Rule-based

techniques make use of a small set of generative rules or grammars to create branches and leaves. Nevertheless, these methods require a high-level of expertise and focus on the creation of synthetic plants with some amount of regularity.

Sketch-based plant modeling. Sketch-based techniques have been introduced mainly in the context of plant modeling, allowing users to create 3D plants and trees using just few strokes. Ijiri *et al.* [IOOI05] present a method for modeling flower petals and leaves, which they later animate [IYKI08] by drawing construction strokes. Okabe *et al.* [OOI07] reconstruct 3D branching skeletons from 2D sketches by maximizing distances between branches. Longay *et al.* [LRBP12] introduce an interactive procedural modeling of trees on a tablet. In [CNX*08] trees are modeled from 2D sketches using probabilistic optimization and parameters obtained from a database of tree models. A global to local sketching approach is applied in [WBCG09] and [IOI06] for trees and flowers, respectively. Nevertheless, these methods do not aim at exact plant reconstruction and are limited on plants with relatively simple geometry and topology.

Image-based plant modeling. A number of approaches try to reconstruct plants from multiple images. Reche-Martinez *et al.* [RMMD04] use registered photos to generate a volumetric representation of the tree canopy and its branches and twigs. Neubert *et al.* [NFD07] improve this by using only loosely arranged input images and a particle-system to produce small branches. Other approaches [SRDT01, TZW*07] extract visual hulls from input images and use L-systems to synthesize branches within these hulls. Later, Tan *et al.* [TFX*08] propose a procedural method that generates a statistically plausible tree model from a single image. Similarly Yan *et al.* [YGCO*14] reconstruct flower petals from an image by fitting geometrical structures while assuming some regular pattern.

Template-based plant modeling. Structure-from-motion was used to compute a 3D point cloud of the plant that later is reconstructed by fitting botanical primitive templates [QTZ*06]. Similar to us, Bradley *et al.* [BNB13] use non-rigid registration to fit a leaf exemplar to fine scale dense foliage. In contrast, we choose not to use a template as our goal is to recover the exact geometry of individual plant leaves and to capture their shape variations. Li *et al.* [LFM*13] present a method for analyzing a growing plant from time-lapse point clouds obtained by a camera projector system. Nevertheless, these approaches cannot recover the full plant geometry and topology as they cannot access self-occluded parts in complex plants.

Point-based plant modeling. With scanning technology becoming available, approaches for plant and tree reconstruction from scan point sets were developed. Xu *et al.* [XGC07] cluster edges in a spanning graph to reconstruct the tree skeleton. Livny *et al.* [LYO*10] reconstruct tree

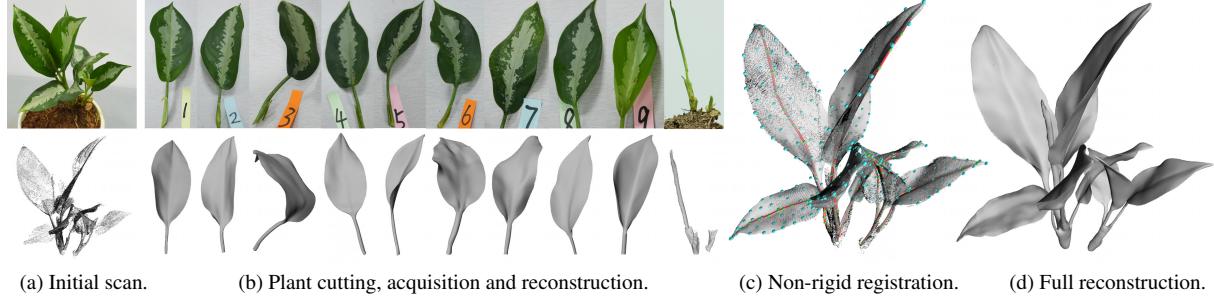


Figure 2: Overview of our intrusive acquisition and plant modeling algorithm.

skeletons from point clouds of multiple trees by computing minimal spanning graphs. In a subsequent work they combine noisy points of the foliage into what they call lobes, thus avoiding reconstructing small details [LPC^{*}11]. Pirk et al. [PSK^{*}12] convert this framework into a dynamic data structure that allows trees to react to their environment.

Bucksch *et al.* [BL08, BLM09] cluster points and form a skeleton by connecting adjacent clusters. Côté *et al.* [CWFV09] synthesize minor geometries on the reconstructed branches based on light scattering properties obtained from additional intensity data. Roumonen *et al.* [RKK^{*}13] reconstruct the branches by locally reconstructing patches that are then combined to the branches. However, none of these techniques attempts at handling the occluded geometry and topology in a principled way.

Recently, Yan *et al.* [YSL^{*}14] introduce a proactive scanning technique, in which the user are allowed to both move around the scanner and push aside the occluding parts so that the hidden areas can be captured. Their motion analysis is however designed to handle only a specific interaction of occluder removal, and hence not suitable for plants with complex self-occlusion patterns. Ijiri *et al.* [IYYI14] develop a X-ray tomography (CT) acquisition system with the same goal of full 3D flowers and plants reconstruction. They use an active contour model to interactively reconstruct the flower shaft and sheet components. In contrast, our method focuses on scanned point clouds and adapts a curve-driven sweeping technique [YHZ^{*}14] for the accurate reconstruction of plant parts in a fully automatic manner.

3. Overview

Our intrusive acquisition process involves cutting the plant into independent parts, thus removing self-occlusions and exposing the full plant geometry to the scanner. Typically cuts are placed near the stem to separate leaves, petals and buds from the complex branching structure.

At initial stage, we loosely scan the plant, yielding a reference geometry as shown in Figure 2(a). This gives us a partial scan, where the interior of the plant may be missing due

to self-occlusions. Nevertheless, the partial scan can guide the subsequent registrations of individual parts.

Next, we scan and reconstruct each plant part independently; see Figure 2(b). We adapt existing point-processing algorithms [HWCO^{*}13, YHZ^{*}14] for computing the ℓ_1 -medial skeleton and the curve-driven reconstruction of thin stem and leaf structures. Due to their delicateness, the dissected stems and leaves may deform and hence have different shapes from those captured in the initial scan. As a result, conventional registration algorithms cannot be applied.

To address this problem, we register each separate plant part back to its physical reference position in a global-to-local manner; see Figure 2(c). We allow globally large non-rigid deformations and locally smooth fitting transformations of each leaf, which account for plant properties and preserve plant features. Throughout the process we avoid plant parts to intersect with formerly registered leaves and stems, and then fuse all model parts into a unified full 3D plant reconstruction as shown in Figure 2(d).

4. Acquisition and plant part reconstruction

4.1. Intrusive acquisition

Due to the nature of plants, such as heavy self-occlusions, inaccessible parts, diverse topologies, slim petioles, and complex foliage geometries, it is impossible to obtain a complete or even semi-complete scan using conventional acquisition. We, first, roughly scan the given plant from several views and register the scans together to form a reference point cloud model, referred as Q , for the whole plant. As shown in Figure 2(a), due to occlusions, the obtained geometric model is incomplete with large missing parts. We then disassemble the plant by cutting it into disjoint leaves with their petioles, from exterior gradually to the interior. Each cut is executed at the root of the stem or the junction of petioles, with the cutting order recorded, e.g., at the top of Figure 2(b).

For each individual part k cut from the plant, an additional scan is performed to capture its shape, free from occlusions. Since most plants have very thin leaves and slim stems, it is

very difficult, if not impossible, to capture manifold meshes for these leaves using current commercial scanners. Hence, we only scan each leaf part from one side, which gives us a thin layer of point cloud Q_k that captures rich geometric details on the leaf surface.

4.2. Sweeping reconstruction for parts

We now reconstruct a manifold mesh M_k for each individual part k based on the scan data Q_k . The conventional parameterization-based [DG06] or implicit function driven surface reconstruction [CBC*01, KBH06] methods are not suitable for such data due to rich geometric features and irregular open boundaries. Inspired by the work of *Morfit (Morph and Fit)* [YHZ*14], we here adopt a curve-sweeping-based reconstruction approach to accurately reconstruct the leaf geometry with feature preservation.

The reconstruction algorithm starts with automatically detecting the two endpoints of the leaf part k : one at the tip of the leaf and the other at the bottom of the petiole. It then computes the ℓ_1 -medial skeleton [HWCO*13] for Q_k with an additional constraint that the skeleton connects the two detected and fixed endpoints. Since the skeleton for the leaf part is in general a single curve, this automatic skeletonization process behaves quite robustly.

Next, the skeleton curve S_k is uniformly sampled into n_k skeletal points $\{p_i\}$ with a spacing defaulted to 4% of its length. For each point p_i , we compute a slicing plane that is perpendicular to the skeleton curve at p_i . Each original point q from Q_k is projected onto its closest slicing plane, forming a set of cross-sectional slices $\{s_i\}$ along the skeleton. Furthermore, based on the directionality degree [HWCO*13] calculated using the projected points in slice s_i , a binary classification similar to the one used by Li et al. [LFM*13] is applied to label the corresponding skeletal point p_i as either *leaf* or *stem* point; see Figure 3(a). In addition, to evaluate whether a point q from the leaf part is close to a boundary, we utilize a feature function $\theta(q)$ similar to the 2D point cloud completeness measure defined in [YHZ*14]. To compute $\theta(q)$, we first project the local neighborhood of q onto the 2D plane calculated by PCA. For each projected point on the plane, we form a unit vector pointing from q to that point. We radially sort these unit vectors and define $\theta(q)$ as the maximum angle between any pair of the neighboring vectors. The function value θ is then normalized into $[0, 1]$. See the color mapping at the top of Figure 3(a), where more red color indicates the point more closely to the boundary.

In general, we find that the stems of plants have cylindrical shapes or can be closely approximated by generalized cylinders. Hence, for each slice s_i associated with a *stem* skeletal point p_i , we fit its profile curve using a circular NURBS curve [WPL06]; see the bottom of Figure 3(b). On the other hand, for each slice s_j associated with a *leaf* skeletal point p_j , except for the one at the tip, we compute a clamped

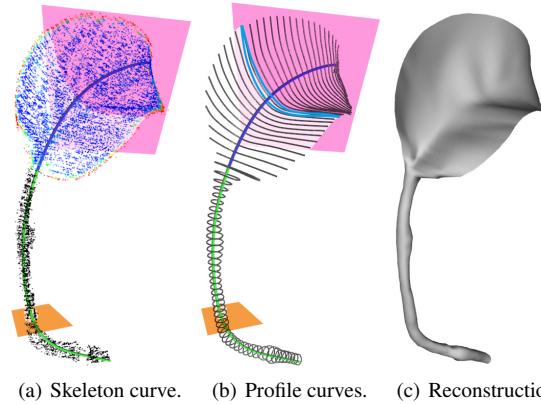


Figure 3: Sweeping reconstruction for a leaf part: (a) the point on skeletal curve is labeled as either leaf (blue) or stem (green) point, and the input scan points belonging to the leaf are colored according to the value of the boundary feature function $\theta(q)$, i.e., more closely to the boundary more red; (b) a clamped (or circular) NURBS profile curve is computed for each leaf (or stem) slice; (c) automatically sweeping the profile curves along the skeleton curve efficiently produces a manifold mesh close to a real leaf part.

NURBS fitting curve. We then generate a closed NURBS profile curve around this clamped NURBS curve with the thickness gradually decay from the center to the boundary; see the top of Figure 3(b). Empirically, we set the center thickness to be 2% of the leaf length. The same number of control points (20 by default) is used for all profile curves, which also converge to the same position at the leaf tip.

With the profile curves $\{c_i\}$ generated for both the leaf and the stem parts, we apply the sweeping reconstruction [YHZ*14] on scan data Q_k to obtain a complete manifold mesh M_k for part k ; see Figure 3(c). That is, we optimize the following function

$$\arg \min_{\{c_i\}} \sum_i (E_d(c_i) + \alpha E_s(c_{i-1}, c_i, c_{i+1}) + \beta_i E_c(c_i)). \quad (1)$$

Comparing with the original Morfit objective function definitions (2-5) in [YHZ*14], our initial profile curves are uniformly and densely sampled and fit the data well, we thus removed the morphing term E_m , resulting only the data fitting term E_d and the smoothness term E_s being used, where the latter one penalizes large deformations between adjacent profile curves with a defaulted weight $\alpha = 0.01$. Meanwhile, we embed the boundary feature function $\theta(q)$ into the data fitting term, which adds stronger weights to the points closer to the leaf boundary and hence encourages the reconstruction to closely follow the scanned boundary. Moreover, to prevent the plant stem from folding during the reconstruction (e.g., see defects in the Morfit result in Figures 7(m-o)),

we add a convex term E_c into the interpolation optimization:

$$E_c(c_i) = 1 - \frac{4\pi \cdot \text{area}(c_i)}{(\text{length}(c_i))^2},$$

where $\text{area}(c_i)$ denotes the area within the profile curve c_i and $\text{length}(c_i)$ is its perimeter. We empirically set the weight $\beta_i = 0.1$ when c_i is at the stem, otherwise, $\beta_i = 0$.

Note that, although the Morfit technique provides users with interactive tools for correcting errors in skeletal and profile curves, these tools are not needed in our system. This is because both skeletal and profile curves can be robustly calculated since we handle only one leaf at a time and utilize the assumptions based on leaf features, i.e., skeleton running through the two endpoints and the profile curves having thin shapes. As a result, our adapted approach can fully automatically reconstruct plant parts through the sweeping optimization while maintaining high reconstruction accuracy; see Figure 7 for one example. If adjacent profile curves intersect each other, the reconstructed surface may contain *folding* artifacts (see e.g., Figure 19 in [YHZ*14]), however, such high curvature cases are really rare for plant leaves.

5. Global-to-local non-rigid registration

With all the individual cut-out parts reconstructed, our next task is to register the part meshes $\{M_k\}$ with the reference point cloud model Q to create a complete mesh model for the original plant. This registration process can also be considered as assembling disjointed meshes $\{M_k\}$ in a coherent but non-right manner using Q as a guide. In practice, we register M_k one by one, in the order that the corresponding parts were cut. Thus, the whole procedure starts with registering outer leaves to well scanned areas, and then gradually moves into the inner and occluded leaves until all meshes are registered.

The registration between M_k and Q cannot be solved using a conventional algorithm due to i) the two are represented using different geometries, i.e., one in a complete mesh and the other in a highly incomplete point cloud; and ii) the same physical part often has different shapes in the two representations due to the deformation caused by cutting off other leaves and the gravity. To properly register them, we developed a novel global-to-local algorithm that can not only accommodate large and non-rigid deformation but also has no strict requirement on the initial placements; see Figure 4. We avoid explicit data segmentation operations during the whole process, such that the registration behaves quite robustly even in challenging cases; see Figure 10.

5.1. Skeleton-driven global registration

The global registration procedure starts with users casually placing the mesh M_k near the desired location. Based on this initial position of the mesh, we compute a dense correspondence set Γ . The set Γ consists of two types of correspondences. One matches each vertex v_i on M_k to its closest

point on Q , whereas the other matches each point q_j from Q to its closest point on M_k . Two points $\langle v_i, q_j \rangle$ are considered as a valid correspondence only when the distance between these two is smaller than a given threshold, which is defaulted to 10% of the plant size.

With Γ defined, we now deform the mesh M_k so that the total inner-correspondence point distance can be minimized. At this global registration step, the deformation for M_k is controlled by translating its individual skeletal point p_i to $p'_i = p_i + t_i$ and/or rotating it about the normal of the corresponding slicing plane by r_i degree. Given a set of t_i and r_i values, the deformed mesh is computed using bounded biharmonic weights introduced in [JBPS11].

To compute optimal t_i and r_i values, we minimize the following objective function:

$$\arg \min_{\{t_i, r_i\}} D(M'_k(\{t_i, r_i\}), Q) + \lambda_g L(S'_k(\{p'_i\})), \quad (2)$$

where $D(\cdot, \cdot)$ is a data fitting term and $M'_k(\{t_i, r_i\})$ refers to the mesh obtained by deforming M_k under transformations $\{t_i, r_i\}$. The regularization term $L(\cdot)$ on the skeleton curve S'_k after the translations $\{t_i\}$ is defined as:

$$L(S'_k) = \frac{1 + \alpha |\Delta S'_k|}{n_k} \sum_{2 \leq i \leq n_k - 1} |p'_{i+1} + p'_{i-1} - 2p'_i|, \quad (3)$$

where $\Delta S'_k$ denotes the change of the skeleton length and n_k is the number of skeletal points sampled on S'_k . This weighted Laplacian regularization can ensure the uniform point distribution along the curve skeleton with the skeleton length preservation. The balancing parameter λ_g in (2) and the weight α in (3) are defaulted to 0.1 and 100, respectively.

The data fitting term $D(\cdot, \cdot)$ computes the weighted sum of the inner-correspondence distances. To cut the computational cost, instead of evaluating the inner-correspondence distances for all correspondences in set Γ , we randomly downsample Γ to a subset Γ' with 1000 correspondence pairs by default, and define $D(\cdot, \cdot)$ as:

$$D(M'_k, Q) = \sum_{\substack{\langle v_i, q_j \rangle \in \Gamma' \\ v'_i \in M'_k, q_j \in Q}} w(v_i, q_j) f(v_i, q_j) |v'_i - q_j|, \quad (4)$$

where $w(\cdot)$ is the weight function and $f(\cdot)$ measures the confidence on the correspondence.

To assign a weight for each correspondence, we consider shape boundaries on both M_k and Q . Detecting boundaries for M_k is straightforward as they correspond to the end points of the NURBS fitting curves. The boundary feature function θ can be utilized again here to evaluate whether a given point q is close to a boundary in the point cloud Q or not. Then, each correspondence $\langle v_i, q_j \rangle$ is given a weight computed using:

$$w(v_i, q_j) = \exp(-\phi(v_i)(1 - \theta(q_j))/\sigma_w^2), \quad (5)$$

where $\phi(v_i)$ denotes the normalized distance between v_i and

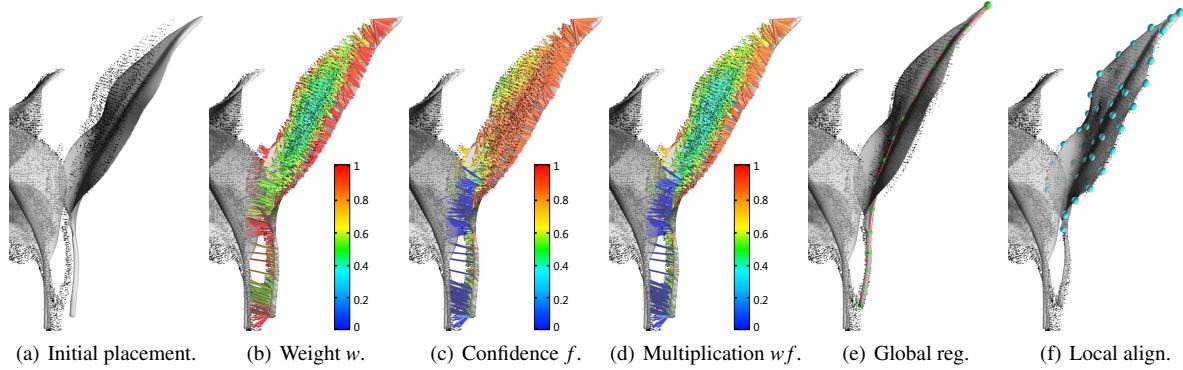


Figure 4: Global-to-local non-rigid registration: (a) the mesh and the corresponding point cloud do not need to be well-aligned in the initial placement; (b) the correspondences between the two are shown in colored lines with color indicating the weight w (red is higher and blue is lower); (c) same correspondences with color indicating the confidence f (the incorrect correspondences along the stem have low (blue) confidence values); (d) multiplying w and f gives good evaluation on the correspondences; (e) through bending and twisting the skeleton, we can register the mesh and the point cloud in a non-rigid manner; (f) differences between the two representations can be further minimized using local skin-guided alignment.

its closest boundary vertex. Assigning higher weights to correspondences closer to boundaries allows the registration to better match the contour of the two shapes; see an example in Figure 4(b). To further evaluate the quality of the correspondences in Γ' , we introduce an additional confidence measure $f(\cdot)$ defined as:

$$f(v_i, q_j) = \exp(-(d(v_i, q_j)/d(\hat{v}_i, q_j))^2/\sigma_f^2), \quad (6)$$

where $d(\cdot, \cdot)$ computes the Euclidean distance between two points, and \hat{v}_i is the closest vertex to the point q_j on all other part meshes except for M_k . Thus, the confidence value is higher only when the current registering mesh M_k is closer to the point than other part meshes; see for example how the confidence values vary from the top to the bottom of the leaf in Figure 4(c).

As a result, the data fitting term weighted with both the boundary preference and the correspondence confidence, e.g., as shown in Figure 4(d), leads to the robust performance in general cases. In our implementation, both Gaussian parameters σ_w and σ_f are set to 1 by default. The deformation parameters $\{t_i, r_i\}$ that optimize (2) are computed using the BFGS solver [NW99]. Once the parameters are found, we update M_k with $M'_k(\{t_i, r_i\})$, S_k with $S'_k(\{t_i\})$, and then update the correspondence set Γ using the new vertex locations. The above skeleton-driven global registration step is then repeated until converges. In practice, we found these process converges after 2-3 iterations; see Figure 4(e).

5.2. Skin-guided local alignment

Even though the above registration step can bend and twist the mesh M_k to best match the point cloud for the same leaf part in Q , it cannot alter the local geometry of the mesh to

align with the scanned data. To further improve the registration and to generate a faithful model, we also apply skin-guided local alignment. That is, instead of deforming M_k through adjusting positions and orientations of its skeletal points, we now manipulate the positions of a set of skin control points down-sampled from the mesh surface.

More precisely, we first uniformly down-sample the vertices on M_k to obtain a set of skin control points ($m_k = 40$ by default) using the WLOP operator [HLZ^{*}09], referred to as $\{u_i\}$ (shown as big blue dots in Figure 4(d)). Through adding a displacement t_i to each skin control point u_i , we can deform the mesh accordingly using bounded biharmonic weights [JBPS11]. Please note that here we do not adjust vertex locations in M_k directly since it may result in a non-manifold mesh. Deforming mesh through bounded biharmonic weights, on the other hand, always maintains the manifold. Also note that increasing the number of skin control points can noticeably improve the alignment accuracy but has higher computation cost.

To compute the optimal displacement parameters, we minimize the following objective function:

$$\operatorname{argmin}_{\{t_i\}} D(M'_k(\{t_i\}), Q) + \lambda_l R(\{u'_i\}), \quad (7)$$

where $D(\cdot, \cdot)$ is the same data fitting term defined in (4) and $M'_k(\{t_i\})$ refers to the mesh deformed under control points displacements of $\{t_i\}$. The regularization term $R(\cdot)$ on the control points $\{u'_i = u_i + t_i\}$ is defined as:

$$R(\{u'_i\}) = \frac{\sum_{1 \leq i \leq m_k} |\Delta_{u'_i}|}{m_k},$$

where $\Delta_{u'_i}$ denotes the average distance change from u'_i to its K-nearest ($K = 4$ by default) neighboring control points.

With R we can ensure the uniform control point distribution about the M_k mesh surface and prevent M_k from undesirable shrinkage. The balancing parameter λ_i is defaulted to 0.1.

This local non-rigid alignment is solved by BFGS and repeated with the updated correspondence Γ until converges, which takes 2-3 iterations. Figure 4(f) shows the final registration result after the skin-guided alignment step, which suggests that the original point cloud is faithfully reconstructed with the mesh model.

5.3. Collision correction

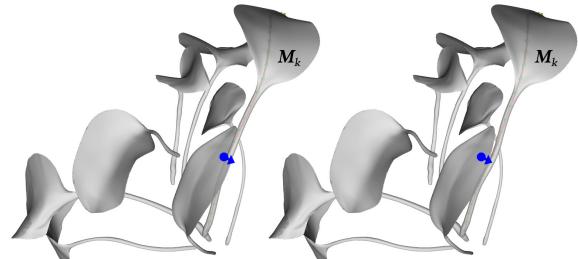
Under ideal situations where the initial acquired reference point cloud model Q contains sufficient data to guide the deformation of individual meshes, simply registering all individual meshes using the aforementioned procedure gives us a complete plant model. However, in many cases, an inner leaf part k may not be well captured in Q due to occlusions. Consequently, the corresponding mesh M_k may not be properly deformed, causing it to collide with other mesh parts.

To address this problem, every time a mesh M_k is registered, we detect if M_k intersects with existing meshes and, if yes, deform M_k to clear the collisions. Here only M_k is changed based on the facts that: i) we always cut the parts from outside gradually to inside and the corresponding meshes are registered in the same order; ii) the scanned data in Q is generally more reliable for the already-registered outer parts than for the to-be-registered inner part M_k .

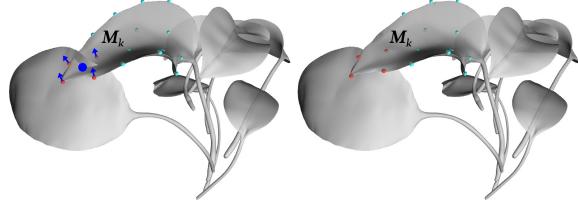
To resolve collision through deformation, we first detect which part of M_k is intersecting with existing meshes and what it is intersecting with. This results in two cases: the stem tip on M_k intersecting with an existing stem, the stem or the leaf on M_k intersecting with an existing mesh. Under the first situation, we skip the collision handling and move to the fusion step to be discussed in Section 5.4.

When the stem part of M_k intersects with an existing mesh M_e , we gradually deform M_k using the skeleton-driven deformation described in Section 5.1 until the collision is cleared. That is, we first calculate the centroid c_{ek} (blue dot in Figures 5(a) and 13(b)) of the area on M_k that is involved in the collision; and compute a displacement vector \mathbf{l}_{ek} that points from c_{ek} to its closest leaf boundary point or stem point on M_e . We then locate the skeletal point p_i on M_k that is closest to c_{ek} and move its position to $p_i + \epsilon \cdot \mathbf{l}_{ek}$, where the constant parameter $\epsilon = 1.05$. Finally, the skeleton-driven deformation with bounded biharmonic weights is applied to deform M_k ; see Figures 5(a) and 13(b). Collision detection is performed again using the newly deformed mesh M'_k and the process is repeated until there is no collision.

When the leaf part of M_k intersects with an existing mesh M_e , we deform M_k using the skin-guided deformation discussed in Section 5.2 to make local adjustment. Similar to the previous case, we first compute the collision centroid c_{ek}



(a) Left: the stem on M_k intersects with an existing mesh. Right: moving the stem to the right (along the blue arrow) through skeleton-driven deformation clears the collision.



(b) Left: the leaf on M_k intersects with an existing mesh. Right: moving the leaf upward (along the blue arrows) through skin-guided deformation clears the collision.

Figure 5: Collision correction for a newly inserted mesh M_k .

(big blue dot in Figure 5(b)) and the corresponding displacement vector \mathbf{l}_{ek} . Then the skin control points (4 small red dots in Figure 5(b)) surrounding c_{ek} are located. The location of each skin control point u_i is moved along its normal \mathbf{n}_i to $u_i + \epsilon(\mathbf{l}_{ek} \cdot \mathbf{n}_i)\mathbf{n}_i$. With these control points moved while keeping others fixed, the mesh M_k is deformed; see Figure 5(b). The collision detection and deformation process is repeated until the collision is fully cleared.

5.4. Stem fusion

The aforementioned registration and collision correct steps place meshes together in a way to best resemble the original plant. However, different meshes may either intersect with or disconnected from each other, so we still do not have a single manifold mesh for the whole plant yet. Hence, at the last step, we perform stem fusion operations.

For plants with trunks, e.g., in Figure 2, the fusion step starts with scanning the remaining trunk after all leaf parts are cut. The obtained point cloud is used to generate a base mesh using Screened Poisson reconstruction [KH13]. For plants that can be completely disassembled into leaf parts, e.g., in Figure 6, the fusion step starts from the most inner part model, i.e., the last piece. Then, one by one, we fuse all the meshes together in the reverse order of the cutting, i.e., from inside to outside. To fuse a mesh M_k , we first detect if it intersects with the mesh that has already been fused together. If it is, merging is performed with a CSG operation [RCG*01]. If

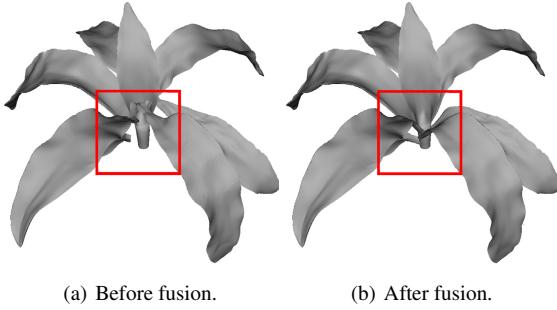


Figure 6: Stem fusion: disconnecting and/or intersecting leaf parts in (a) are merged together to form a manifold mesh for the whole plant.

| #Fig | 10.1 | 10.2 | 10.3 | 10.4 | 8 | 9 |
|-------|------|------|------|------|-----|-----|
| N_Q | 80K | 82K | 67K | 27K | 89K | 72K |
| N_L | 17 | 13 | 13 | 17 | 8 | 9 |
| N_S | 15 | 17 | 18 | 16 | 17 | 15 |
| T_S | 157s | 110s | 126s | 108s | 98s | 75s |
| N_C | 40 | 40 | 40 | 40 | 40 | 40 |
| T_C | 170s | 156s | 148s | 100s | 84s | 87s |
| T_P | 4 | 2 | 1 | 2 | 3 | 0 |

Table 1: Timing for our registration and reconstruction of six different plants shown in Figures 10, 8 and 9. N_Q : number of initial scan points; N_L : number of leaves; N_S : number of average skeletal points per leaf; T_S : time used for skeleton-driven global registration; N_C : number of average control points per leaf; T_C : time used for skin-guided local alignment; T_P : time used for the post-processing.

no intersection is found, we search for the closest stem or the closest point on the ground plane, onto which we snap the bottom of M_k through skeleton-driven deformation, before the CSG merging. As shown in the results, this yields manifold meshes for the various plants.

6. Results and discussion

Scanning and timing. We use the Artec Spider scanner for scanning all examples presented in this paper. The Spider's best working distance is [0.17m, 0.3m] with the resolution around 1mm and the angular field of view $H30^\circ \times W21^\circ$. The average sweeping reconstruction time per leaf is about 3 seconds, and the average vertex number per reconstructed leaf mesh is 2K. In Table 1, we list the data information and the running time for the registration and the final reconstruction of six plants. The computation timing is measured on an Intel Xeon E5-2687W CPU @3.40GHz with 16GB RAM.

Quantitative evaluations. We start our experiments with a synthetic plant model so that the ground truth is available.

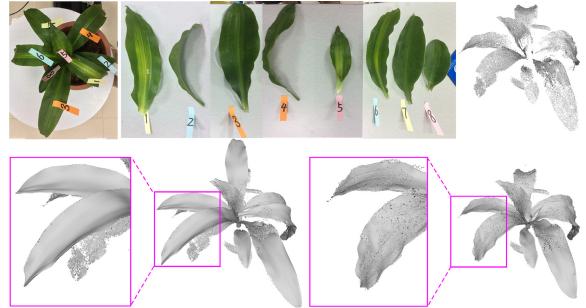


Figure 8: The reconstruction for a *Dracaena sanderiana* whose leaves (top middle) are highly curved and are not developable surfaces. Even though the reference point cloud captured (top right) is incomplete, casually placing the individual mesh model over the point cloud (bottom left) is sufficient for proper registration (bottom right).

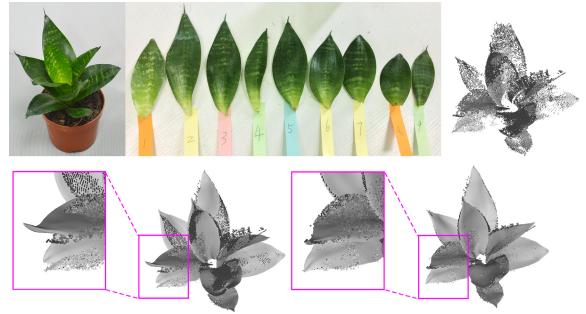


Figure 9: The reconstruction for *Sansevieria trifasciata* whose leaves are tightly pack together. Without cutting off the outer leaves, it is close to impossible to scan the shape of inner ones. However, even though reference point cloud model (top right) contains large missing areas, our registration approach is still able to deform individual meshes to match the scan reasonably well (bottom right) with even loosely placed initial positions (bottom left).

Due to strong self-occlusions, the virtual scan for the whole plant as shown in Figure 7(b) is not complete enough for automatic and accurate reconstruction. For example the Ball-pivoting surface reconstruction [BMR*99] approach failed to produce a connected and manifold mesh; see Figure 7(c).

To evaluate our approach, we detach individual leaves from the model as if they were cut from a real plant. Each leaf part is then deformed using a non-rigid physical-based deformation technique [BJ05], which simulates the leaf shape changes after the gravity orientation is altered. The deformed models (gray ones in Figure 7(d)) are scanned virtually from one side to generate individual point clouds; see Figure 7(e). The leaf models that we automatically reconstruct from the point clouds and the corresponding reconstruction errors are

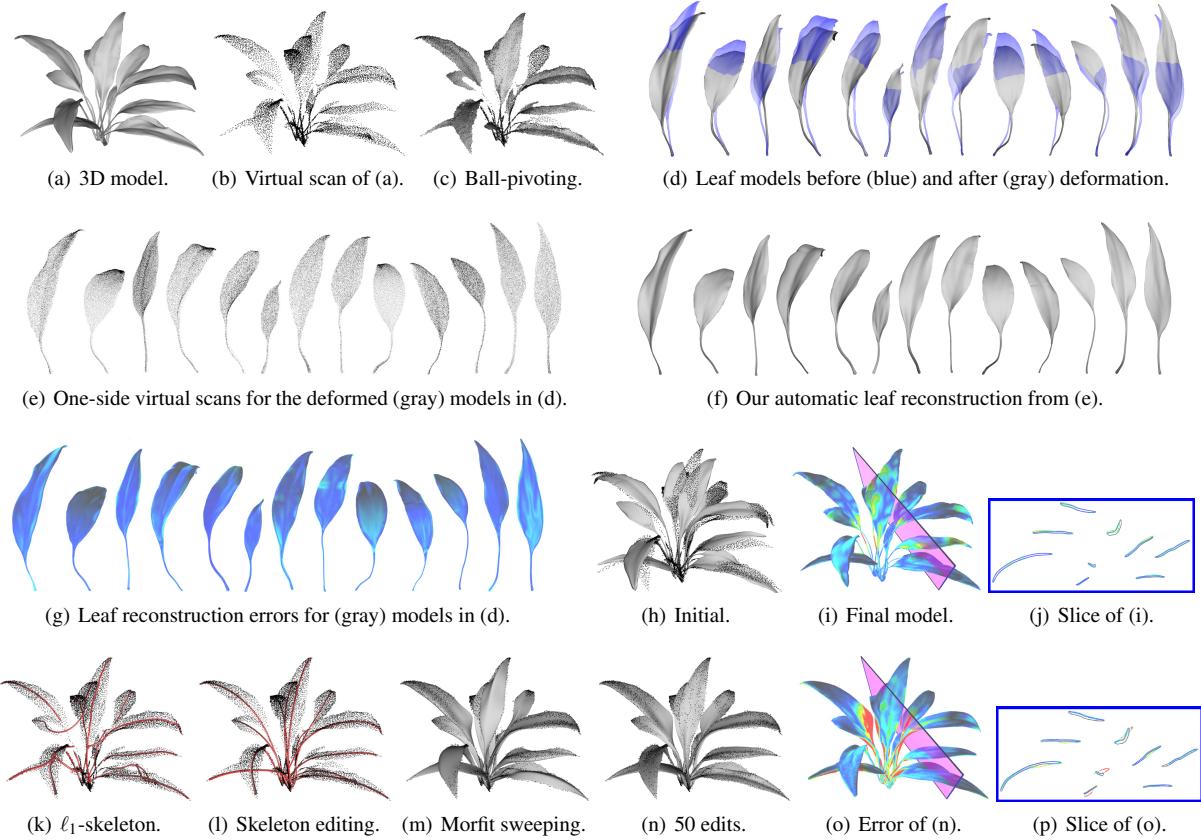


Figure 7: Quantitative evaluation on a synthetic (*Spathiphyllum kochii*) model (a) for both our intrusive plant reconstruction (e-i) and the interactive Morfit reconstruction (k-o). Our approach yields the more accurate reconstruction (i-j), whereas the non-intrusive Morfit approach has over-smoothed output and higher reconstruction error (n-p) even after interactively editing 50 profile curves. The maximum reconstruction errors shown in (g), (i), and (o) are about 0.007, 0.015 and 0.036, respectively, given that the input modal (a) has been normalized into a unit cube.

shown in Figures 7(f) and (g), respectively. Roughly positioning these leaf models into the whole-plant scan gives us an initial plant model (Figures 7(h)), which evolves into the final model after global-to-local registration and fusion (Figures 7(i)). It is worth noting that our approach successfully aligns the reconstruction models of deformed leaves with the original undeformed leaves only based on the incomplete point clouds of the whole plant as shown in Figure 7(b).

In comparison, to reconstruct a model for the same plant using the Morfit algorithm, we first need to extract a skeleton (Figure 7(k)) from the whole-plant scan and manually correct errors in the skeleton (Figure 7(l)). Then, after a mesh model is obtained through Morfit sweeping, additional manually editing on the profile curves are needed to adjust the shapes and positions of the leaves. Nevertheless, even after interactively editing 50 profile curves, the reconstruction model is still over-smoothed and has higher error than the

one obtained using our approach; compare Figures 7(i) and (o) with their zoomed cross slices in Figures 7(j) and (p).

Tests on real plants. We also applied the presented intrusive acquisition technique on a variety of real plants. They demonstrate how well the technique handles a number of challenging situations. For example, the *Aglaonema crispum* plant shown in Figure 1 (top) contains soft inner leaves whose shapes are constrained by the outer ones. Once the outer leaves are removed, the shapes of inner ones change. The leaves of the *Dracaena sanderiana* var. *mediopicta* shown in Figures 1 (bottom) and 8 have highly curved and non-developable surfaces, making both the reconstruction for individual leaves and their registration very challenging. The *Epipremnum aureum* and *Aglaonema modestum* shown in Figures 5 and 10 (3rd row) have long and slim stems, which often require careful collision handling. The *Sansevieria trifasciata* in Figure 9, the *Stromanthe san-*

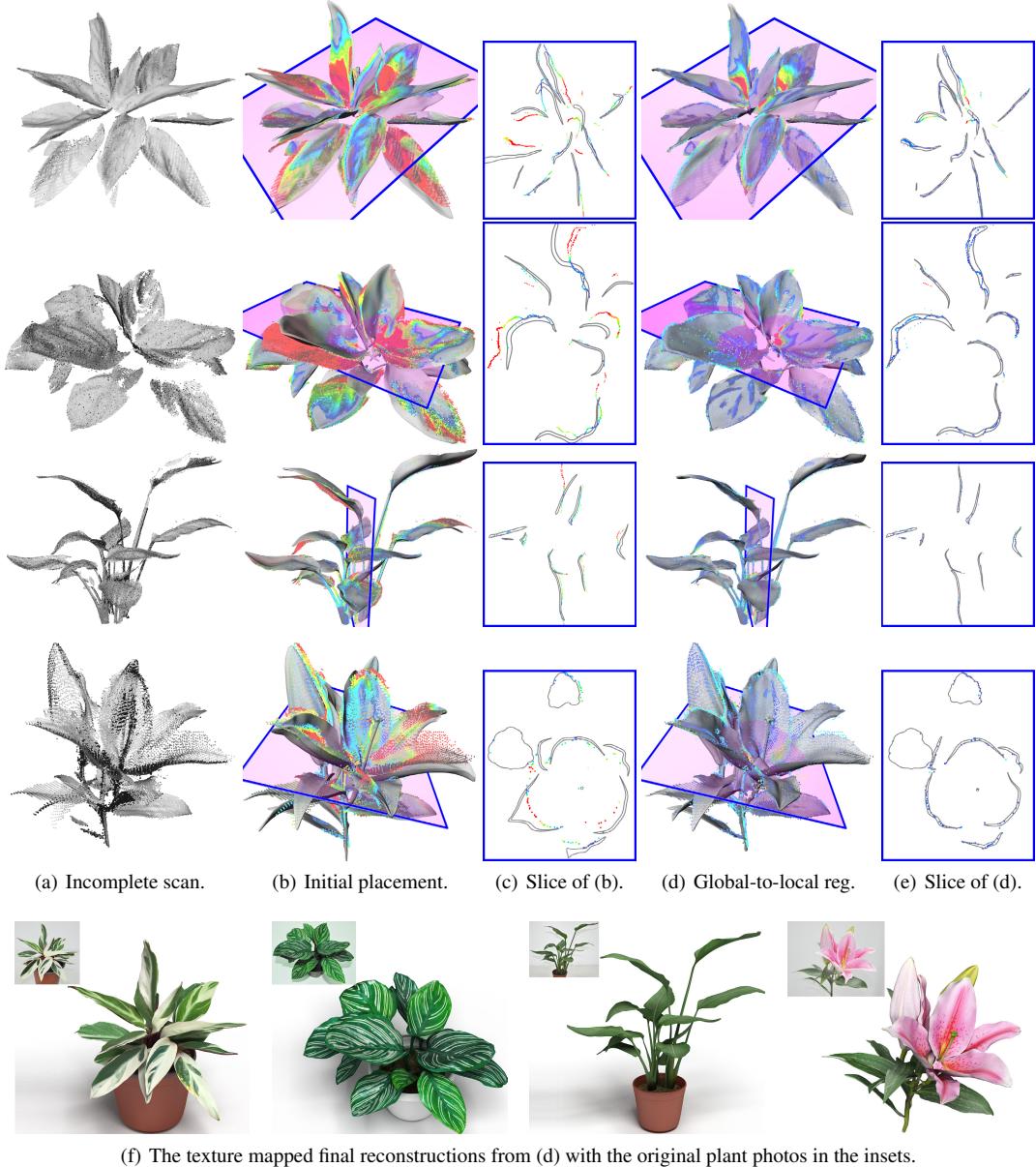


Figure 10: Reconstruction of four complex plants. From top to bottom, we have *Stromanthe sanguinea*, *Calathea majestica*, *Aglaonema modestum*, and *Lilium casa blanca*. The results show that the presented global-to-local registration step (d) can effectively reduce the reconstruction error caused by the initial imprecise positioning (b) of the leaf models. The color of scan points in (b) and (d) encodes the closest distance to the meshes (red is further and blue is closer). For better illustration, the effect of registration is also demonstrated using selected 2D cross sections in (c) and (e). The texture (from plant photos) mapped models in (d) are rendered in (f), which faithfully resemble the plant's appearances in the real photos (insets).

guinea and Calathea majestica in Figure 10 (first 2 rows) have leaves tightly packed, making it nearly impossible to scan the inner ones without using an intrusive approach. The results show that our method performs well in all cases.

Although the algorithm is designed for scanning plants, it can also be applied to flowers with large and overlaid petals. Figure 10 (4th row) shows our results on a *Lilium casa blanca*. Although the petals are highly curved and non-

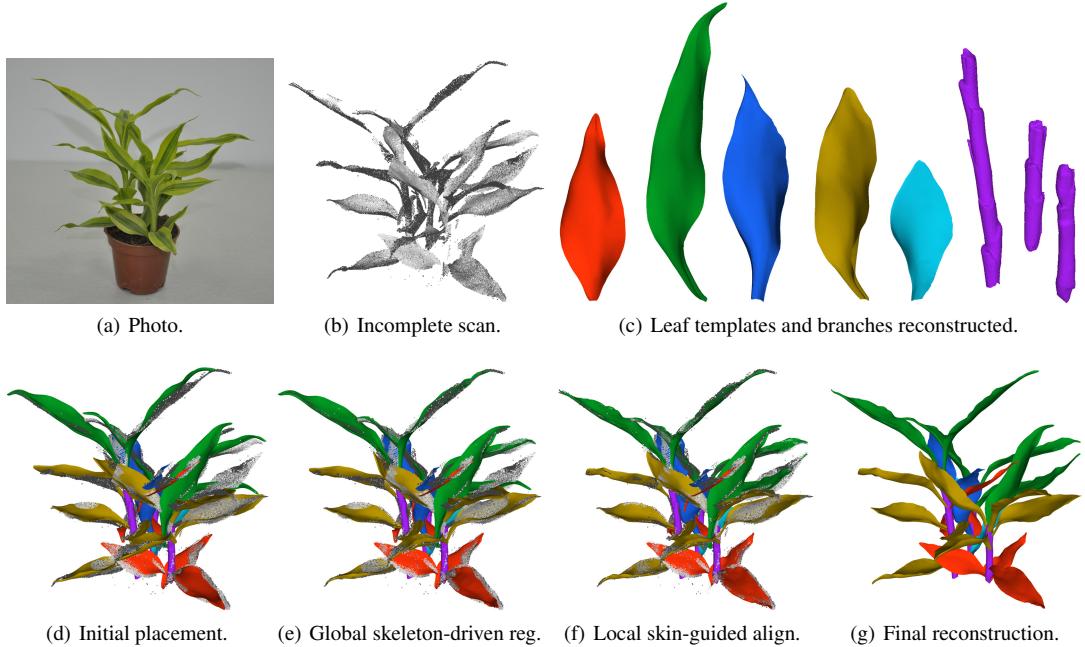


Figure 11: The reconstruction of a *Dracaena sanderiana* var. *marginatum* (a) using leaf templates. After scanning five selected leaves as templates (c), the whole plant can be reconstructed by reusing the templates at multiple places. As shown in (e-f), the global-to-local registration step can effectively adjust the template positions and shapes to fit individual leaves.

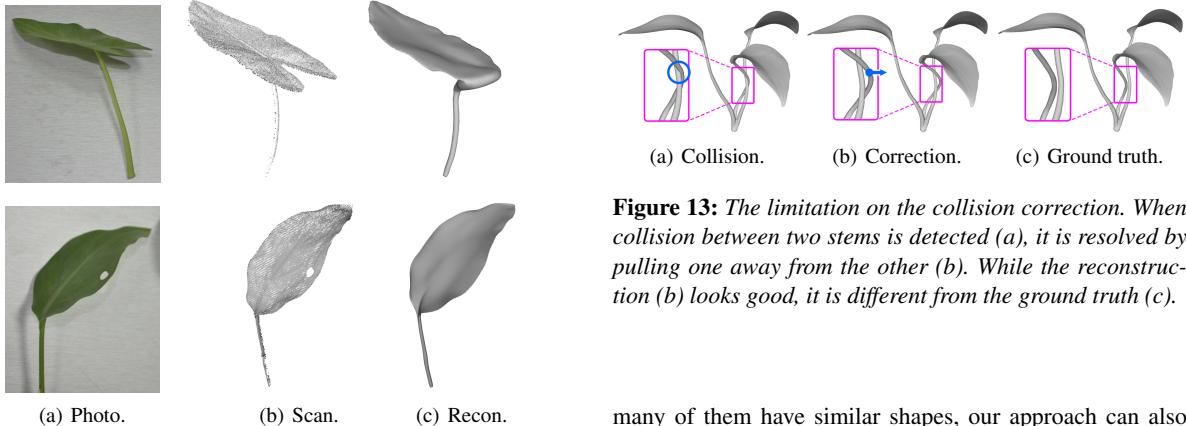


Figure 12: The limitation on the leaf reconstruction through sweeping. Top row: we assume the leaf has tips on both ends and thus cannot handle leaves with concave shapes very well. Bottom row: original holes on the leaf surface will be ignored due to the sweeping reconstruction operation.

developable, our approach is able to accurately reconstruct their shapes and register them into a complete flower model.

Finally, for plants that have a large number of leaves but

many of them have similar shapes, our approach can also reconstruct complete models using just a few templates obtained by scanning representative leaves. Figure 11 shows how a *Dracaena sanderiana* var. *marginatum* plant with 25 leaves is reconstructed from only five leaf templates.

Limitations. Our technique still has its limitations. First of all, the plant to be scanned need to be cut into pieces. Hence, users may not want to apply this technique on rare or valuable plants. In fact, we did attempt to scan inner leaf parts by moving away the outer ones, rather than cutting them, similar to the way that proactive scanning [YSL*14] does. However, we found that, due to the limited access angles for

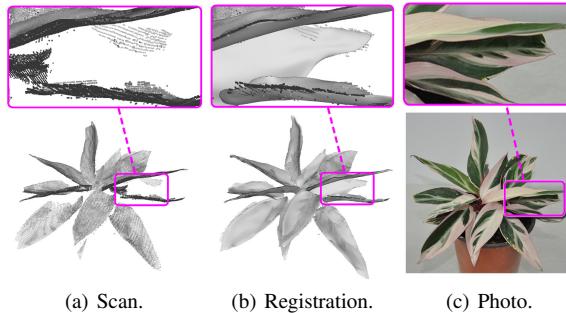


Figure 14: The limitation on handling heavily occluded leaves, where the initial scan data is missing significantly (a). Although the mesh model for the leaf highlighted in the pink box is aligned well to the partially scanned point cloud (b), its final position and shape differ from the original one (ground truth) shown in the photo (c).

the scanner and the usually large non-rigid leaf deformations, the quality of the scan data and the following registration are not good enough for accurate and automatic reconstruction. We plan to further study the feasibility of an intrusive but non-destructive acquisition approach.

At technical level, our current approach assumes that the plant leaves have a clear tip point, which can be automatically identified and used as a constraint for the skeletonization and the subsequent sweeping reconstruction. For plants with round-shaped or heart-shaped (concave) leaves, e.g., in Figure 12 (top row), such a constraint needs to be elevated. Figure 12 (bottom row) also shows that holes on the leaf surface are ignored since the sweeping reconstruction we used assumes that there is no topology changes among the profile curves on different slices. In addition, our approach resolves collision through pulling the colliding parts away from each other. Under certain scenarios, this may pull the parts toward the wrong directions; see Figure 13. And the presented collision detection and solving algorithm may not converge when multiple complicate collisions occur; for instance, many petals are packed in a small area. Additional user interventions are then needed. Finally, when a leaf is hardly visible in the whole-plant scan, our approach cannot precisely register the corresponding leaf model, due to the lack of reliable matching correspondences; see Figure 14.

7. Conclusion and future work

This paper presents a novel intrusive acquisition approach that can capture highly deformable and self-occluded plants. The results show that the full geometries and topologies of a variety of plants can be faithfully reconstructed.

Future work. While we focused on plant scanning in this paper, the global-to-local skeleton-to-skin non-rigid regis-

tion approach that we developed is likely applicable to other articulate yet soft objects. For example, a possible future direction is to conduct complete scans for animals when they are put into sleep to obtain accurate 3D models and then register the models onto the dynamic but incomplete scans obtained in real-time when the animals perform actions. This would provide us an accurate and continuous 4D model.

Besides to develop an intrusive but non-destructive acquisition approach aforementioned, another possible direction we like to explore is to replace the manually initial placement with a quality-driven position searching scheme, making the whole registration and reconstruction process fully automatic. If this is achievable, the proposed technique can be utilized more easily and its applications can further broaden, such as 3D SLAM in a complex environment.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive comments. This work was supported in part by NSFC (61379090, 61202224, 61202221), National 973 Program (2014CB360503), Shenzhen Innovation Program (CXB201104220029A, JCYJ20140901003939034, JCYJ20140901003938994), SIAT Innovation Program for Excellent Young Researchers (201305, 201402), NSERC (293127) and the Israel Science Foundation.

References

- [BJ05] BARBIČ J., JAMES D. L.: Real-time subspace integration for st. venant-kirchhoff deformable models. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 24, 3 (2005), 982–990. 8
- [BL08] BUCKSCH A., LINDBERGH R.: Campino – a skeletonization method for point cloud processing. *ISPRS J. photogrammetry and remote sensing* 63, 1 (2008), 115–127. 3
- [BLM09] BUCKSCH A., LINDBERGH R., MENENTI M.: Skeltre - fast skeletonisation for imperfect point cloud data of botanic trees. In *Proc. Eurographics Workshop on 3D Object Retrieval* (2009), pp. 13–27. 3
- [BMR*99] BERNARDINI F., MITTLEMAN J., RUSHMEIER H., SILVA C., TAUBIN G.: The ball-pivoting algorithm for surface reconstruction. *IEEE Trans. Visualization & Computer Graphics* 5, 4 (1999), 349–359. 8
- [BNB13] BRADLEY D., NOWROUZEZHRAI D., BEARDSLEY P.: Image-based reconstruction and synthesis of dense foliage. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* (2013), 74:1–74:10. 2
- [CBC*01] CARR J. C., BEATSON R. K., CHERRIE J. B., MITCHELL T. J., FRIGHT W. R., MCCALLUM B. C., EVANS T. R.: Reconstruction and representation of 3D objects with radial basis functions. *Proc. of SIGGRAPH* (2001), 67–76. 4
- [CNX*08] CHEN X., NEUBERT B., XU Y.-Q., DEUSSEN O., KANG S. B.: Sketch-based tree modeling using markov random field. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 27, 5 (2008), 109:1–109:9. 2
- [CWFV09] CÔTÉ J.-F., WIDLAWSKI J.-L., FOURNIER R. A., VERSTRAETE M. M.: The structural and radiative consistency of three-dimensional tree reconstructions from terrestrial lidar. *Remote Sensing of Environment* 113, 5 (2009), 1067–1081. 3

- [DG06] DEY T. K., GOSWAMI S.: Provable surface reconstruction from noisy samples. *Comput. Geom. Theory Appl.* 35, 1 (2006), 124–141. 4
- [HLZ^{*}09] HUANG H., LI D., ZHANG H., ASCHER U., COHEN-OR D.: Consolidation of unorganized point clouds for surface reconstruction. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 28, 5 (2009), 176:1–176:7. 6
- [HWCO^{*}13] HUANG H., WU S., COHEN-OR D., GONG M., ZHANG H., LI G., CHEN B.: ℓ_1 -medial skeleton of point cloud. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 32 (2013), 65:1–65:8. 3, 4
- [IOI06] IJIRI T., OWADA S., IGARASHI T.: Seamless integration of initial sketching and subsequent detail editing in flower modeling. In *Computer Graphics Forum* (2006), vol. 25, pp. 617–624. 2
- [IOOI05] IJIRI T., OWADA S., OKABE M., IGARASHI T.: Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 24 (2005), 720–726. 1, 2
- [IYKI08] IJIRI T., YOKOO M., KAWABATA S., IGARASHI T.: Surface-based growth simulation for opening flowers. In *Proc. of Graphics Interface* (2008), pp. 227–234. 2
- [IYYI14] IJIRI T., YOSHIZAWA S., YOKOTA H., IGARASHI T.: Flower modeling via x-ray computed tomography. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 33 (2014), 48:1–48:10. 2, 3
- [JBPS11] JACOBSON A., BARAN I., POPOVIĆ J., SORKINE O.: Bounded biharmonic weights for real-time deformation. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 30, 4 (2011), 78:1–78:8. 5, 6
- [KBH06] KAZHDAN M., BOLITHO M., HOPPE H.: Poisson surface reconstruction. *Proc. Eurographics Symp. on Geometry Processing* (2006), 61–70. 4
- [KH13] KAZHDAN M., HOPPE H.: Screened poisson surface reconstruction. *ACM Trans. on Graphics* 32, 3 (2013), 29:1–29:13. 7
- [LFM^{*}13] LI Y., FAN X., MITRA N. J., CHAMOVITZ D., COHEN-OR D., CHEN B.: Analyzing growing plants from 4d point cloud data. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 32, 6 (2013), 157:1–157:10. 2, 4
- [LPC^{*}11] LIVNY Y., PIRK S., CHENG Z., YAN F., DEUSSEN O., COHEN-OR D., CHEN B.: Texture-lobes for tree modelling. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 30, 4 (2011), 53:1–53:10. 3
- [LRBP12] LONGAY S., RUNIONS A., BOUDON F., PRUSINKIEWICZ P.: Treesketch: interactive procedural modeling of trees on a tablet. In *Proc. Int. Symp. on Sketch-based Interfaces and Modeling* (2012), pp. 107–120. 1, 2
- [LYO^{*}10] LIVNY Y., YAN F., OLSON M., CHEN B., ZHANG H., EL-SANA J.: Automatic reconstruction of tree skeletal structures from point clouds. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 29, 6 (2010), 151:1–151:8. 2
- [NFD07] NEUBERT B., FRANKEN T., DEUSSEN O.: Approximate image-based tree-modeling using particle flows. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 26, 3 (2007), 71:1–71:8. 2
- [NW99] NOCEDAL J., WRIGHT S.: *Numerical Optimization*. Springer, New York, 1999. 6
- [OOI07] OKABE M., OWADA S., IGARASHI T.: Interactive design of botanical trees using freehand sketches and example-based editing. In *ACM SIGGRAPH Courses* (2007). 2
- [PHL^{*}09] PALUBICKI W., HOREL K., LONGAY S., RUNIONS A., LANE B., MĚCH R., PRUSINKIEWICZ P.: Self-organizing tree models for image synthesis. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 28 (2009), 58:1–58:10. 1
- [PJM94] PRUSINKIEWICZ P., JAMES M., MĚCH R.: Synthetic topiary. In *Proc. of SIGGRAPH* (1994), pp. 351–358. 2
- [PL96] PRUSINKIEWICZ P., LINDENMAYER A.: *The Algorithmic Beauty of Plants (The Virtual Laboratory)*. Springer, 1996. 1, 2
- [PSK^{*}12] PIRK S., STAVA O., KRATT J., SAID M. A. M., NEUBERT B., MĚCH R., BENES B., DEUSSEN O.: Plastic trees: interactive self-adapting botanical tree models. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 31, 4 (2012), 50:1–50:10. 3
- [QTZ^{*}06] QUAN L., TAN P., ZENG G., YUAN L., WANG J., KANG S. B.: Image-based plant modeling. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 25 (2006), 599–604. 1, 2
- [RCG^{*}01] ROCCHINI C., CIGNONI P., GANOVELLI F., MONTANI C., PINGI P., SCOPIGNO R.: Marching intersections: an efficient resampling algorithm for surface management. *Proc. IEEE Int. Conf. on Shape Modeling & Applications* (2001), 296–305. 7
- [RKK^{*}13] RAUMONEN P., KAASALAINEN M., KAASALAINEN S., KAARTINEN H., VASTARANTA M., HOLOPAINEN M., DISNEY M., LEWIS P., ET AL.: Fast automatic precision tree models from terrestrial laser scanner data. *Remote Sensing* 5 (2013), 491–520. 3
- [RMMD04] RECHE-MARTINEZ A., MARTIN I., DRETTAKIS G.: Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 23, 3 (2004), 720–727. 2
- [SRDT01] SHLYAKHTER I., ROZENOER M., DORSEY J., TELLER S.: Reconstructing 3D tree models from instrumented photographs. *IEEE Computer Graphics and Applications* 21, 3 (2001), 53–61. 2
- [TFX^{*}08] TAN P., FANG T., XIAO J., ZHAO P., QUAN L.: Single image tree modeling. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 27, 5 (2008), 108:1–108:7. 2
- [TZW^{*}07] TAN P., ZENG G., WANG J., KANG S. B., QUAN L.: Image-based tree modeling. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 26 (2007), 87:1–87:6. 2
- [WBCG09] WITHER J., BOUDON F., CANI M.-P., GODIN C.: Structure from silhouettes: a new paradigm for fast sketch-based design of trees. In *Computer Graphics Forum* (2009), vol. 28, pp. 541–550. 2
- [WPL06] WANG W., POTTMANN H., LIU Y.: Fitting b-spline curves to point clouds by curvature-based squared distance minimization. *ACM Trans. on Graphics* 25, 2 (2006), 214–238. 4
- [XGC07] XU H., GOSSETT N., CHEN B.: Knowledge and heuristic-based modeling of laser-scanned trees. *ACM Trans. on Graphics* 26, 4 (2007), 19:1–19:13. 2
- [YGC0^{*}14] YAN F., GONG M., COHEN-OR D., DEUSSEN O., CHEN B.: Flower reconstruction from a single photo. *Computer Graphics Forum (Proc. of Eurographics)* 33, 2 (2014). 1, 2
- [YHZ^{*}14] YIN K., HUANG H., ZHANG H., GONG M., COHEN-OR D., CHEN B.: Morfit: Interactive surface reconstruction from incomplete point clouds with curve-driven topology and geometry control. *ACM Trans. on Graphics (Proc. of SIGGRAPH Asia)* 33, 6 (2014), 41:1–41:12. 3, 4, 5
- [YSL^{*}14] YAN F., SHARF A., LIN W., HUANG H., CHEN B.: Proactive 3d scanning of inaccessible parts. *ACM Trans. on Graphics (Proc. of SIGGRAPH)* 33, 4 (2014), 157:1–157:8. 3, 11