

Flower Modeling via X-ray Computed Tomography

Takashi Ijiri¹

Shin Yoshizawa¹

Hideo Yokota¹

Takeo Igarashi²

¹RIKEN

²The University of Tokyo

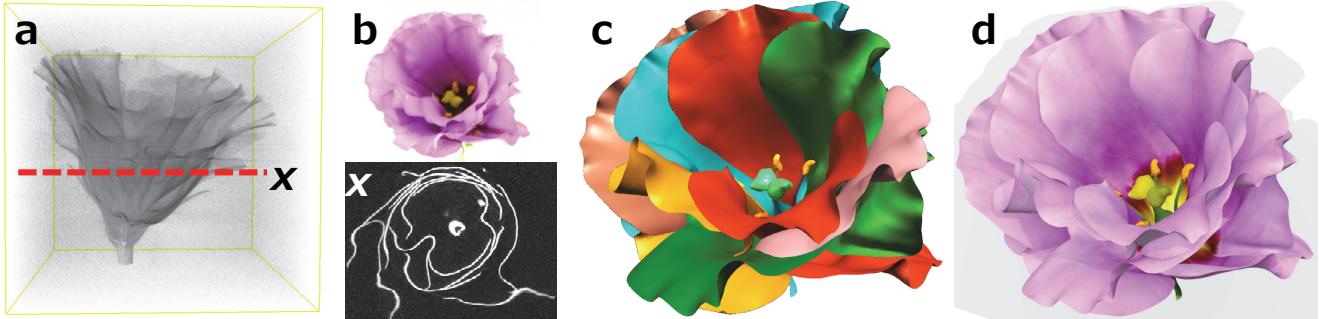


Figure 1: *Eustoma* model generated with our technique. From a CT volume (a) of a sample flower (b: photograph), we reconstructed a flower model (c). Pane (x) is a cross section of the volume (a). The reconstructed flower model was rendered with texture (d).

Abstract

This paper presents a novel three dimensional (3D) flower modeling technique that utilizes an X-ray computed tomography (CT) system and real-world flowers. Although a CT system provides volume data that captures the internal structures of flowers, it is difficult to accurately segment them into regions of particular organs and model them as smooth surfaces because a flower consists of thin organs that contact one another. We thus introduce a semi-automatic modeling technique that is based on a new active contour model with energy functionals designed for flower CT. Our key idea is to approximate flower components by two important primitives, a shaft and a sheet. Based on our active contour model, we also provide novel user interfaces and a numerical scheme to fit these primitives so as to reconstruct realistic thin flower organs efficiently. To demonstrate the feasibility of our technique, we provide various flower models reconstructed from CT volumes.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques; I.4.6 [Image Processing]: Segmentation—Edge and feature detection.

Keywords: Flower modeling, X-ray CT, active contours.

Links: [DL](#) [PDF](#)

1 Introduction

Plant modeling has been an important and challenging problem in the graphics community. Various interesting methods, including procedural modeling [Prusinkiewicz and Lindenmayer 1990; Palubicki et al. 2009], image-based modeling [Quan et al. 2006], and

sketch-based modeling [Ijiri et al. 2005; Anastacio et al. 2009], have been presented and have achieved some realistic tree and plant models. However, it remains difficult to model complex flowers, because flowers usually contain highly occluded structures and many thin organs contacting one another.

One effective way to model flowers is by reconstruction from real-world samples. The goal of this paper is to present a reconstruction technique for realistic and complex flower models. We use an X-ray computed tomography (CT) system that provides grayscale volumes whose intensity corresponds to X-ray attenuation rates and that can clearly capture internal structures of flowers (Fig. 1a). Anatomically, a flower consists of six types of organs: stem, receptacle, pistil, stamen, petal, and sepal [Glimn-Lacy and Kaufman 2006]. Given a flower CT volume, we construct surface models of these organs. For two reasons, however, this is a computationally challenging problem. One is that a flower consists of many thin organs with similar CT intensities, and the other is that a flower has many sheet-like organs (e.g., petals) that contact one another. Generally, it is difficult to segment such regions when using conventional approaches [Adams and Bischof 1994; Boykov and Jolly 2001].

In this paper, we propose a novel semi-automatic modeling technique based on a new active contour model consisting of active curves and surfaces. We approximate flower components by two key primitives, a shaft and a sheet, and fit them to the CT volume with our active curve and surface models. When fitting a shaft, we compute its axis by using the active curve. When fitting a sheet, we first compute a boundary curve network consisting of the active curves and then interpolate the network by using the active surface. We design active curve and surface energies specific to flower organs and present an efficient numerical scheme for our primitive fitting based on a shortest path algorithm and gradient descent optimization. We also provide a set of user interfaces for modeling each of the six flower organs intuitively with our active curves and surfaces.

Our implementation works in real-time, and it fits the primitives immediately when the user specifies control points. Fig. 1 shows a *Eustoma* model, generated by our technique in 30 min interaction, in which occluded internal organs and wavy petals were properly reconstructed. Because the flower models are reconstructed from real-world samples, they are potentially suitable for botanical observations as well as for use in realistic animations and simulations.

2 Related work

Procedural plant modeling. The best-known procedural modeling method for plants is the L-system [Lindenmayer 1968; Prusinkiewicz and Lindenmayer 1990]. This has been extended to simulate a wide variety of plant growth phenomena, such as plant interactions with their environments [Prusinkiewicz et al. 1994; Měch and Prusinkiewicz 1996], gradual variation in certain features along an axis [Prusinkiewicz et al. 2001], and self-organized activation of buds [Palubicki et al. 2009]. With well-tuned growth rules, the L-system produces highly realistic plant models. Lintermann and Deussen [1999] presented a graph-editing interface to support intuitive procedural modeling. The main focus of these methods, however, is to model the branching structures of plants.

Sketch-based plant modeling. The sketch-based approach is well suited to interactive plant modeling because it allows the user to specify various curved features of plants by simply drawing strokes. Okabe et al. [2005] presented an algorithm to generate 3D trunks and branches of trees from 2D sketch strokes. Longay et al. [2012] applied a space colonization algorithm to sketch-based tree modeling. Ijiri et al. [2005] and Anastacio et al. [2009] developed methods for modeling flower petals (or leaves) by drawing construction strokes. However, these methods are limited to relatively simple petal shapes and are not easily employed to generate wavy shapes, like those seen in Fig. 1.

Image-based plant modeling. Recently, some researchers have used image-based techniques for tree and plant modeling. Tan et al. [2008] generated a tree model from a single image. They grew a branching structure such that its 2D projection was fitted to the input image and its 3D shape was fairly balanced. Quan et al. [2006] and Bradley et al. [2013] proposed image-based tree modeling methods for bushes. They reconstructed 3D point clouds from multiple photographs of a plant using a structure-from-motion method and then fitted primitive leaf models to the point clouds. Zhang et al. [2014] applied a similar method to flower petals. Li et al. [2013] presented a method for analyzing a growing plant from time-lapse point clouds obtained by a camera projector system. Image-based approaches, however, are difficult to apply to flower modeling, as flowers usually have occluded organs, and an image-based approach does not reconstruct invisible objects.

Computed tomography in graphics. CT systems have been employed in the graphics field to model realistic objects. Reche et al. [2004] introduced a volumetric rendering method for trees from photographs. Trifonov et al. [2006] developed a volume reconstruction system for transparent objects such as glasses by removing refraction optically. Gregson et al. [2012] proposed a stochastic algorithm to capture 3D images of turbulent fluids. Zhao et al. [2011] used an X-ray CT system to develop volumetric appearance models of fabrics. Also, X-ray CT devices have become readily accessible and have been used for botanical analyses (e.g., flower structures [Stuppy et al. 2003]). However, previous works in botany do not provide computational methods for modeling flower surfaces.

Curve and surface fitting. There are many useful approaches to fitting curves and surfaces to point clouds. Here, we briefly review geometric methods related to our technique. Medial axis [Blum 1967] is one of the best-known skeletal representations and is often employed to reconstruct curves [Abeysinghe and Ju 2009; Huang et al. 2013] and surfaces [Amenta et al. 1998]. To incorporate CT information, we adapt the idea of *centeredness* [Abeysinghe and Ju 2009] to our shaft and sheet primitive fitting. Our sheet primitive fitting is related to curve network interpolations, the typical strategy for which consists of mesh refinement and surface fairing; see [Moreton and Séquin 1992; Nealen et al. 2007; Bessmeltsev et al. 2012] and references therein. On the other hand, these pure geomet-

ric approaches, including traditional splines (e.g., Bézier patches, subdivision surfaces), do not fit their surfaces to certain volume regions. Implicit functions are also popular for shape reconstructions (e.g., [Süßmuth and Greiner 2007; Kazhdan and Hoppe 2013]). To our knowledge, only Ijiri et al. [2013] explicitly provided an implicit function that interpolates both curve networks and volume edges. Unfortunately, this method is not suitable for modeling wavy, occluded, or sheet-like surfaces.

Active contours. The *active contour model*, also called snake, was proposed by Kass et al. [1988]. It segments the image region by minimizing an internal energy representing boundary smoothness and an external energy representing image edge consistency. Osher and Sethian [1988] introduced its level set formulation. These methods have various applications and have been studied widely in the graphics and vision communities [Sethian 1999]. Cohen and Cohen [1993] extended the active contour to a surface model for 3D volume segmentation. Caselles et al. [1997] introduced geodesic active contours which lay in a Riemannian space defined by image contents. Active surface models are also useful for shape modeling, such as surface reconstruction [Sharf et al. 2006] and mesh registration [Eckstein et al. 2007]. The topic of active contours is a popular research field, and many studies have been published in recent decades. Unfortunately, conventional active contours are not readily usable for flower CT volumes in which thin surfaces contact one another, and it is not trivial to extend them for our purpose.

3 Variational Model for Flower CT

This section establishes fundamental formulations of our active curves and surfaces for CT-based flower modeling. The next section formulates specific energy functionals for individual flower organs and presents a set of user interfaces for providing the necessary constraints. Section 5 introduces an efficient numerical scheme for fitting the active curves and surfaces.

According to flower anatomists [Glimn-Lacy and Kaufman 2006], a flower mainly consists of six organs: stem, receptacle, pistil, stamen, petal, and sepal (Fig. 2a). When we observe real flower CT volumes, even those of different species, a specific organ has similar shapes and structures. Furthermore, most of the organs consist of shaft- or sheet-like shapes. A stamen filament, for example, has a shaft-like shape (see Fig. 2G), and a petal has a sheet-like shape (see Fig. 2H). We therefore assume in this paper that the six organs consist of two types of key primitives: *shaft* and *sheet*. A shaft is a curved cylinder whose radius varies along its axis, and a sheet is a surface homeomorphic to an open disk with adaptive thickness. We model the shafts and sheets by using active curves and surfaces.

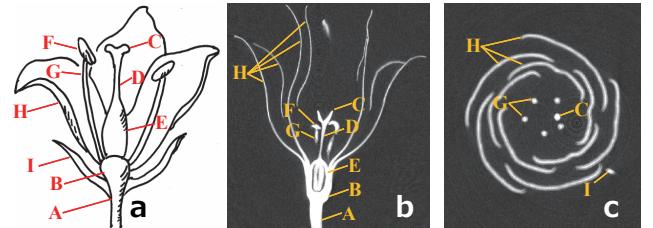


Figure 2: A flower structure (a) and vertical/horizontal cut planes (b, c) of a flower CT volume. A flower has a stem (A); receptacle (B); pistils consisting of stigma (C) style (D) and ovary (E); stamens consisting of anther (F) and filament (G); petals (H); and sepals (I).

3.1 Active Curves and Surfaces

Active curve model. We introduce an active curve to model the shaft and a curve network for constraining the sheet. Consider a

curve $\mathbf{C} = \mathbf{C}(t) \in \mathbb{R}^3$ where its parameter t lives in a connected domain $\Omega_c \subset \mathbb{R}$. We define our active curve energy by

$$E_c = \int_{\Omega_c} \frac{1}{2} |\mathbf{C}''(t)|^2 + \alpha |\mathbf{C}'(t)^T \mathcal{M}(\mathbf{C}(t)) \mathbf{C}'(t)| dt \quad (1)$$

where $\mathbf{C}' = \frac{d\mathbf{C}}{dt}$, $\mathbf{C}'' = \frac{d^2\mathbf{C}}{dt^2}$, a positive definite matrix $\mathcal{M} = \mathcal{M}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ specifies the local Riemannian metric at a point $\mathbf{x} \in \mathbb{R}^3$, and $\alpha > 0$ is a parameter to balance the two terms. The first and second terms correspond to the internal and external energies of the active contour [Kass et al. 1988] with extension of \mathcal{M} . The first term evaluates the smoothness of the curve; it is larger where the curve bends rapidly. The second term presents the curve length in a Riemannian space defined by \mathcal{M} . Minimizing Eq. (1) gives us our active curve. Our active curve is controlled by the metric \mathcal{M} ; we describe different metrics depending on flower organs in Section 4.

Active surface model. Computing surfaces that fit certain volume regions is also indispensable for our flower modeling because petals and sepals have sheet-like shapes. Consider a surface $\mathbf{S} = \mathbf{S}(u, v) \in \mathbb{R}^3$ with its closed boundary curve $\partial\mathbf{S}$ where its parameters (u, v) live in a connected domain $\Omega_s \subset \mathbb{R}^2$. We define our active surface energy by

$$E_s = \iint_{\Omega_s} \frac{1}{2} (\mathbf{S}_{uu}^2 + 2\mathbf{S}_{uv}^2 + \mathbf{S}_{vv}^2) + \beta |\mathcal{B}\mathbf{S}_u \times \mathcal{B}\mathbf{S}_v| dudv \quad (2)$$

where $\mathbf{S}_u = \frac{\partial \mathbf{S}}{\partial u}$, $\mathbf{S}_v = \frac{\partial \mathbf{S}}{\partial v}$, $\mathbf{S}_{uu} = \frac{\partial^2 \mathbf{S}}{\partial u^2}$, $\mathbf{S}_{uv} = \frac{\partial^2 \mathbf{S}}{\partial u \partial v}$, $\mathbf{S}_{vv} = \frac{\partial^2 \mathbf{S}}{\partial v^2}$, $\mathcal{B} = \mathcal{B}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ is a matrix at a point \mathbf{x} , and $\beta > 0$ is a parameter that balances the effects of the two terms. The first term corresponds to the thin plate energy of the active surface [Cohen and Cohen 1993] and evaluates the smoothness of the surface. The second term corresponds to surface area in a Riemannian space defined by the metric $\mathcal{B}^T \mathcal{B}$. Minimizing Eq. (2) gives us our active surface. Our active surface is controlled by the matrix \mathcal{B} , and we define it so as to fit the surface to petal and sepal regions in the volume (see Section 4.3).

The second terms in Eqs. (1) and (2) formulate curve and surface energies by using arc-length and surface-area in the volume-dependent Riemannian spaces, respectively. This idea was inspired by the geodesic active contours [Caselles et al. 1997], which showed equivalence between the computation of active contours and that of geodesics in a data-dependent Riemannian space. In contrast to previous active contour models, our technical contributions include interactive primitive fitting techniques and new energy models for flower CT volumes.

3.2 Basic Tensor Metric for Flower CT

For defining \mathcal{M} and \mathcal{B} in Eqs. (1) and (2), let us first consider an image manifold $\mathbf{r} = \mathbf{r}(\mathbf{x}) \in \mathbb{R}^4$ and its Jacobian $\mathbf{J} = \mathbf{J}(\mathbf{x}) \in \mathbb{R}^{4 \times 3}$ [Sochen et al. 1998]:

$$\mathbf{r}(\mathbf{x}) = \begin{pmatrix} \mathbf{x} \\ I(\mathbf{x}) \end{pmatrix}, \quad \mathbf{J}(\mathbf{x}) = \begin{pmatrix} \mathbf{I}_3 \\ \nabla I(\mathbf{x})^T \end{pmatrix} \quad (3)$$

where $I = I(\mathbf{x})$ is the volume intensity at $\mathbf{x} \in \mathbb{R}^3$ and $\mathbf{I}_3 \in \mathbb{R}^{3 \times 3}$ is an identity matrix. This Jacobian \mathbf{J} has geometric meaning: if we use $\mathcal{M} = \mathbf{J}^T \mathbf{J}$, the second term in Eq. (1) becomes the arc-length of \mathbf{C} on the image manifold (Riemannian covariant metric tensor). Moreover, if we employ \mathbf{J} to \mathcal{B} with a wedge product for the second term in Eq. (2), this term leads to the hyper-surface area of \mathbf{S} on the image manifold.

Let us construct a tensor field $\mathbf{T}(\mathbf{x}) \in \mathbb{R}^{3 \times 3}$ over the volume by

$$\mathbf{T}(\mathbf{x}) = G_\sigma * (\mathbf{J}(\mathbf{x})^T \mathbf{J}(\mathbf{x})) \quad (4)$$

where $G_\sigma *$ is a Gaussian convolution with a standard deviation σ . Eq. (4) is a smooth version of the metric tensor of the image manifold \mathbf{r} where it consists of the coefficients of the first fundamental form of \mathbf{r} ; the eigenanalysis of $\mathbf{J}^T \mathbf{J}$ provides stretch (local metric distortion) magnitudes and their corresponding orthogonal directions of \mathbf{r} . Let us denote the corresponding eigenvalues and eigenvectors of $\mathbf{T}(\mathbf{x})$ by $(\lambda_1, \lambda_2, \lambda_3)$ and $(\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3)$ where $\lambda_1 \geq \lambda_2 \geq \lambda_3$, and an orthogonal frame consisting of \mathbf{e}_i by $\mathbf{U} = (\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3) \in \mathbb{R}^{3 \times 3}$.

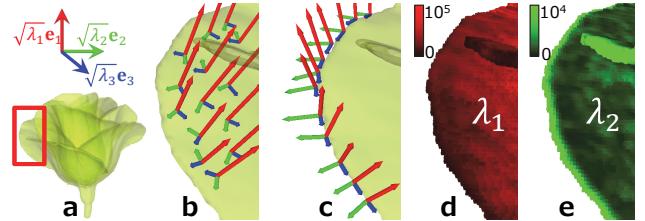


Figure 3: An eigenanalysis of $\mathbf{T}(\mathbf{x})$ in Eq. (4). CT volume is binarized by simple thresholding (a). We visualize the eigensystems on a petal region in (b) and around the petal edge region in (c). In (d) and (e), λ_1 and λ_2 are visualized in red and green, respectively.

Roughly speaking, \mathbf{e}_1 gives a local axis along which the volume intensity varies most strongly. Also, \mathbf{e}_2 and \mathbf{e}_3 are the second and third orthogonal axes of intensity variation. Note that \mathbf{e}_1 is not usually parallel to ∇I because of convolving G_σ , although the dominant eigenvector of $\mathbf{J}^T \mathbf{J} = \mathbf{I}_3 + \nabla I \nabla I^T$ is parallel to ∇I with its corresponding eigenvalue $(1 + |\nabla I|^2)$. Here, ∇I evaluates a direction corresponding to a maximum rate of intensity variation, whereas \mathbf{e}_1 measures a direction of a maximum distortion (total intensity variation around \mathbf{x}). Fig. 3 illustrates the eigenanalysis of an actual flower CT volume. On a petal region that has a sheet-like shape (Fig. 3b), \mathbf{e}_1 coincides with a normal vector of the petal sheet. Also, \mathbf{e}_2 and \mathbf{e}_3 are tangent to the sheet, but they are not aligned in certain directions because λ_2 is close to λ_3 (i.e., \mathbf{T} is close to $\mathbf{J}^T \mathbf{J}$ in such regions). Around the edge of a petal region (i.e., boundary of the petal sheet in Fig. 3c), \mathbf{e}_1 is parallel to the petal sheet normal because total intensity variation around the petal edge is still highest along the petal sheet normal. Because intensity variation is usually lowest along the sheet boundary curve, \mathbf{e}_2 and \mathbf{e}_3 are aligned with respect to the sheet boundary; \mathbf{e}_2 is tangent to the sheet boundary, and \mathbf{e}_3 is tangent to the sheet boundary curve (Fig. 3c). Also, λ_1 is smaller and λ_2 is larger near the petal edge (Fig. 3d,e). Using these observations on the basic tensor $\mathbf{T}(\mathbf{x})$, we define \mathcal{M} and \mathcal{B} in Eqs. (1) and (2) in Section 4.

4 Organ-by-organ Flower Modeling

A flower has a repetitive structure consisting of multiple organs with similar, but slightly different, shapes. To support efficient modeling of such a structure, it is important to provide user interfaces specialized to particular components [Ijiri et al. 2005]. In our technique, we represent the six flower organs (Fig. 2) with the key primitives (shaft and sheet) that consist of our active curves and surfaces. We then introduce user interfaces specifically tailored for particular organs to intuitively provide the constraints necessary for fitting the primitives.

The overall modeling procedure is shown in Fig. 4. In the beginning, we extract foreground voxels V_f that are intersections between simple thresholding and textural binarization [Sauvola and Pietikäinen 2000] (we set its parameters by $k = 0.2$ with 5 voxel radius) results. The threshold is selected by the user to cover a whole flower, including little air or noise (Fig. 4a). In our technique, we classify the six organs into three groups based on their structural

similarities, and the user constructs organ models sequentially in the following order: stem/receptacle, pistil/stamen, and petal/sepal (Fig. 4b-d).

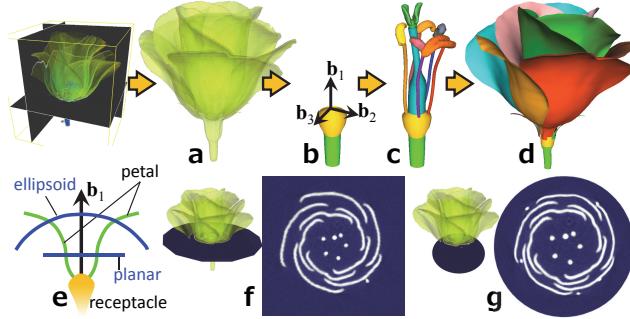


Figure 4: Overall modeling procedure and cross sections.

In the modeling process, control points are specified using cross sections of the volume. The placement of cross sections is therefore important for efficient interaction. After stem/receptacle modeling (Fig. 4b), we introduce two cross sections to enhance the subsequent pistil/stamen and petal/sepal modeling procedures (Fig. 4c,d). We first construct a local coordinate system ($\mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3$) at the top of the receptacle (Fig. 4b), with \mathbf{b}_1 parallel to the receptacle axis and with \mathbf{b}_2 and \mathbf{b}_3 orthogonal to \mathbf{b}_1 and to each other. We then provide two types of cross sections (Fig. 4e): a *planar cross section* perpendicular to \mathbf{b}_1 (Fig. 4f) and an *ellipsoidal cross section* with its center placed at the top of the receptacle (Fig. 4g). In our observation, a pistil or stamen often appears as a circular spot, and a petal or sepal usually appears as a single curve in the cross sections. Thus, with these cross sections, the user readily recognizes flower organs and intuitively specifies necessary control points. Providing the two types of cross sections is important to cover various organs. The planar cross section is useful for treating organs that are aligned along \mathbf{b}_1 , and the ellipsoidal cross section is useful for treating bent petals or radially arranged organs. These two cross sections are used in the pistil/stamen and petal/sepal modeling processes; the user can always switch them, move the planar cross section along \mathbf{b}_1 , and modify the ellipsoid radius by mouse manipulations.

4.1 Stem and Receptacle Modeling

Stem and receptacle model. The stem and receptacle are approximated by a single shaft with a top hemi-ellipsoid. Our active curve serves as the shaft axis (Fig. 5a). The shaft is divided into a lower stem part and an upper receptacle part (Fig. 5d).

Modeling process. To model this organ group, the user specifies several control points $\{\mathbf{q}_k^{\text{shaft}} \in \mathbb{R}^3\}$ where the shaft axis (i.e., the active curve) passes through. The control points are used as constraints in Eq. (1). To specify a control point, the user places a cross section where the organ is observed well and clicks on the cross section (Fig. 5b,c). We provide a cut stroke interface [Ijiri et al. 2013] in which a stroke drawn on the screen is swept in the view-depth direction to generate a curved cross section. The control points $\{\mathbf{q}_k^{\text{shaft}}\}$ are always sorted according to their height along the stem. When at least two control points are placed, our technique generates a shaft whose axis passes through the control points and fits the medial axis of V_f (blue curve in Fig. 5d). The ratio for dividing the shaft into stem and receptacle parts and the radius of the top hemi-ellipsoid are tuned interactively with slider bars so that the resulting model is fitted to the input flower CT.

Primitive formulation. We formulate the active curve of the shaft axis that passes through $\{\mathbf{q}_k^{\text{shaft}}\}$ by minimizing Eq. (1) with the

following metric:

$$\mathcal{M}^{\text{shaft}}(\mathbf{x}) = \frac{\mathbf{I}_3}{1 + d^b(\mathbf{x})} \quad (5)$$

where $d^b(\mathbf{x})$ is a discrete distance field in the foreground voxels V_f from the binarized CT boundary; i.e., $d^b(\mathbf{x}) = 0$ at the boundary voxels of V_f . With this metric, a curve that passes nearer the medial axis of V_f has smaller energy in Eq. (1), resulting in the desired shaft axis for the stem and receptacle model.

We next compute the radius of the shaft. At a point $\mathbf{C}(t)$ on the active curve \mathbf{C} , we construct multiple rays from $\mathbf{C}(t)$ such that they are orthogonal to the tangent direction of the active curve and are arranged radially with a constant angle interval (Fig. 5e). The rays are extended until they reach the background voxels or a user-specified maximum length which is easily estimated from the CT volume. The shaft radius at $\mathbf{C}(t)$ is defined as the average length of the rays. From the active curve and the radius, a shaft representing the stem and receptacle is generated (Fig. 5d).

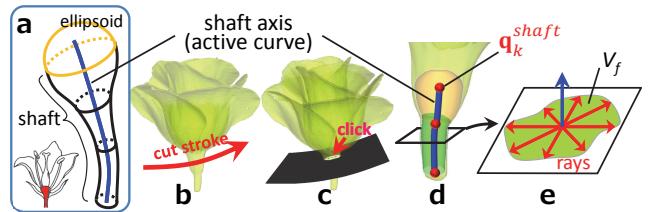


Figure 5: Stem and receptacle modeling. The model of this organ group consists of a shaft with a hemi-ellipsoid (a). A cross section is placed by drawing a cut stroke (b), and a control point is specified on it (c). A shaft generated from control points (red points) is divided into stem (green) and receptacle (yellow) parts (d). The shaft radius is computed by shooting multiple rays (e).

4.2 Pistil and Stamen Modeling

Pistil and stamen model. Since pistils and stamens have similar structures, they are formed from the same set of primitives: a shaft and a *head* (Fig. 6a). The shaft approximates the ovary and style of a pistil or the filament of a stamen. The head is a free-form watertight surface representing the stigma of a pistil or the anther of a stamen. Unfortunately, it is difficult to model the head with specific primitives because of wide variation.

Modeling process. A flower usually has multiple pistils and stamens, and we model them one by one. For modeling the shaft part, we provide the same user interface and formulation used in the stem and receptacle modeling (Section 4.1). To model the head, the user places seeds in the target region (stigma or anther) by clicking on the cross section.

Primitive formulation. The shaft part is computed similarly to the stem and receptacle. The only difference is that the control points are sorted according to their distance from the receptacle; we compute a discrete distance field $d^r(\cdot)$ in V_f by setting $d^r(\mathbf{x}) = 0$ at positions \mathbf{x} of all voxels touching the receptacle and sort the control points according to $d^r(\cdot)$. The head part is computed by seeded region growing [Adams and Bischof 1994]. We grow a voxel region from the seeds within a user-specified intensity threshold and convert the region boundary to a surface model (Fig. 6).

4.3 Petal and Sepal Modeling

Petal and sepal model. Petals and sepals are represented with a

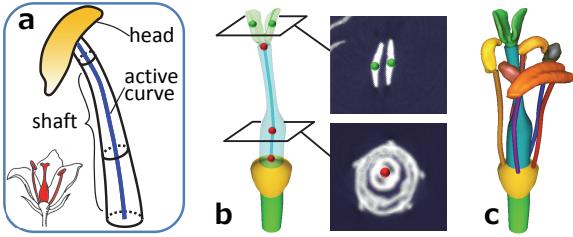


Figure 6: Pistil and stamen modeling. This organ group consists of a shaft and a head (a). The model is generated from the control points (red points) and head seeds (green points), specified by clicking on the cross section (b). (c) provides an example where we model a pistil and five stamens of a Eustoma.

sheet primitive. As illustrated in Fig. 7a, the sheet primitive consists of two types of active curves (a *beam curve* and a *boundary curve*) that form a curve network and one active surface that interpolates the curve network. At a point $\mathbf{S}(u, v)$ on the active surface \mathbf{S} , two scalar functions $\phi_f(u, v)$ and $\phi_b(u, v)$, i.e., front and back offsets, are defined to represent the thickness.

Modeling process. A flower usually has multiple petals and sepals, and we model them one by one. To model a petal sheet, the user specifies multiple beam curves on differently positioned cross sections. A petal appears as a single curve on the cross sections (Fig. 7b), and the beam curve is placed along this curve. To specify a beam curve, the user sequentially clicks on the cross section, as indicated by the arrows in Fig. 7b. Then, our technique places control points $\{\mathbf{q}_k^{beam} \in \mathbb{R}^3\}$ at the clicked points and optimizes a beam curve that passes through $\{\mathbf{q}_k^{beam}\}$ and fits the target petal (Fig. 7c) by minimizing Eq. (1) with the metric given in Eq. (6).

After beam curve specification, our technique generates the sheet primitive automatically: we compute the boundary curve by minimizing Eq. (1) with the metric given in Eq. (8) and then optimize the active surface that interpolates the curve network and fits the target petal region by minimizing Eq. (2) with the matrix given in Eq. (9) (Fig. 7d-g). Once the sheet primitive is generated, it is rasterized as a binary volume and visualized on the cross section, which helps subsequent modeling of the rest of petals (see our supplemental video for a demonstration).

In our experience, three or four beam curves are sufficient for correctly reconstructing petals with simple geometric configurations, and slightly more beam curves are necessary for petals in contact with many other petals. Also, we do not allow placement of a beam curve that intersects one already placed.

Primitive formulation. To obtain a sheet primitive that satisfies the user inputs and fits the target petal region in the volume, we first construct a curve network consisting of beam and boundary curves and then interpolate the curve network with the active surface.

Beam curve. A beam curve \mathbf{C} passing through the user-clicked points $\{\mathbf{q}_k^{beam}\}$ is computed by minimizing Eq. (1) with the metric \mathcal{M}^{beam} . In a flower CT volume, a petal usually has higher intensity at its center. To obtain a beam curve that traverses the petal center, we define the metric by

$$\mathcal{M}^{beam}(\mathbf{x}) = \mathbf{U} D(\mathbf{x}) \text{diag}(1, a, a) \mathbf{U}^T \quad (6)$$

where \mathbf{U} is the orthogonal frame consisting of eigenvectors \mathbf{e}_i of $\mathbf{T}(\mathbf{x})$ at a point \mathbf{x} given by Eq. (4), $\text{diag}(\cdot, \cdot, \cdot)$ is a diagonal matrix, $a \in [0, 1]$ is a parameter (we set $a = 0.1$), and the centeredness $D(\cdot) \in [0, 1]$ is defined as follows [Abeysinghe and Ju 2009]:

$$D(\mathbf{x}) = \frac{I_{\max}(\mathbf{x}) - I(\mathbf{x})}{I_{\max}(\mathbf{x}) - I_{\min}(\mathbf{x})} \quad (7)$$

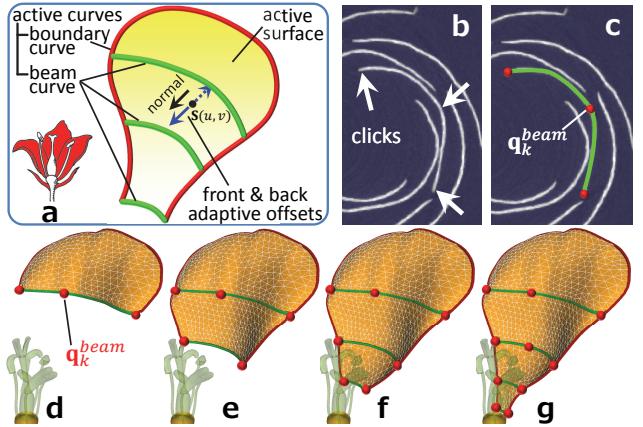


Figure 7: Petal and sepal modeling. This organ group is represented with a sheet primitive (a). A beam curve is specified by clicking on the cross section (b,c). A sheet is generated from multiple beam curves (d-g) with their corresponding boundary curve.

where $I_{\max}(\mathbf{x})$ and $I_{\min}(\mathbf{x})$ are the maximum and minimum intensities, respectively, in the local window centered around \mathbf{x} . $D(\mathbf{x})$ returns a smaller value when the volume intensity at \mathbf{x} is higher than those at neighboring points. With this metric, a curve following a higher intensity region and running more orthogonally to \mathbf{e}_1 has lower energy, resulting in the desired beam curve. In our current implementation, we set the window radius at five voxels so that the window always contains petal regions and background in terms of our CT devices.

Boundary curve. To provide a constraint for fitting the sheet, we compute its boundary curve \mathbf{C} automatically from the beam curves. The boundary curve should pass through the end points of all beam curves. We first reorder all the end points to obtain control points $\{\mathbf{q}_k^{bound} \in \mathbb{R}^3\}$; as shown in Fig. 8(a,b), all beam curves are oriented in the clockwise direction with respect to \mathbf{b}_1 , and their end points are sorted using the distance $d^r(\cdot)$ from the receptacle.

We compute a boundary curve \mathbf{C} passing through $\{\mathbf{q}_k^{bound}\}$ by minimizing Eq. (1) with the metric

$$\mathcal{M}^{bound}(\mathbf{x}) = \mathbf{U} \frac{\lambda_1}{\lambda_2 \lambda_3} \text{diag}(\lambda_1, \lambda_2, \lambda_3) \mathbf{U}^T \quad (8)$$

where \mathbf{U} and λ_i are given by Eq. (4). With this metric, a curve that is placed nearer the edge of a petal region has lower energy because λ_1 is smaller and λ_2 is larger near the edge (Fig. 3). Also, a curve along the petal edge has lower energy because \mathbf{e}_3 is aligned along the petal edge and $\lambda_1 \geq \lambda_2 \geq \lambda_3$. As a result, this metric provides an active curve that follows the edge of a petal region (Fig. 8c).

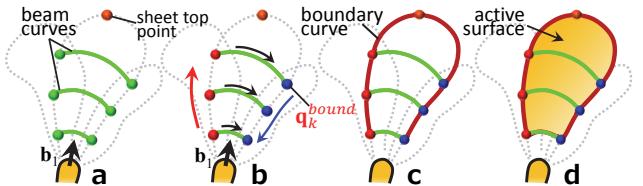


Figure 8: Curve network for fitting the sheet primitive.

Active surface. Given a curve network consisting of the beam and boundary curves (Fig. 8c), we compute a surface $\mathbf{S} \in \mathbb{R}^3$ that interpolates the curve network, as shown in Fig. 8d, by minimizing Eq. (2) with the matrix

$$\mathcal{B}(\mathbf{x}) = D(\mathbf{x}) \text{diag}(1, a, a) \mathbf{U}^T \quad (9)$$

where a and $D(\cdot)$ are given by Eqs. (6) and (7), respectively. With this matrix, a surface that traverses a higher-intensity region has lower energy (see Eq. (7): $D(\cdot)$). Also, a surface with a normal close to e_1 has lower energy. Thus, this metric results in a desired active surface located on the center of the target petal region.

After obtaining the surface, we estimate its thickness, i.e., the front and back offsets $\phi_f(u, v)$ and $\phi_b(u, v)$. At a point $S(u, v)$ on the surface S , we calculate the front and back rays; the front ray directs in the normal vector of $S(u, v)$ and the back ray orients in the opposite direction. The rays are extended until they reach the background voxels or a user-specified maximum length. Then, $\phi_f(u, v)$ and $\phi_b(u, v)$ are defined with the lengths of corresponding rays at $S(u, v)$. We inflate the active surface by the front and back offsets to generate a thick petal model. The accuracy of the estimated thickness depends on the input volume resolution. If necessary, a few iterations of mesh smoothing (e.g., [Kobbelt et al. 1998]) to the obtained thick petal model provide a visually pleasing surface.

Additional treatment. Since flowers have various shapes, additional heuristic treatments are sometimes necessary. When a petal or sepal has a sharp edge, our boundary curve may be too smooth at the top of the sheet. To trace a sharp petal edge correctly, we allow the user to optionally place points around the sheet top by clicking on the boundary surface of V_f (Figs. 8 and 9). We use these sheet-top points as additional constraints in $\{q_k^{\text{bound}}\}$ for our boundary curve computation. When a target flower has complex structure around the receptacle (e.g., Rosa) or a target petal does not have a sheet-like shape around the receptacle (e.g., Aristoeumeria), it is better to set a simpler metric for our active curve and surface adaptively. We use $\mathcal{M}(\mathbf{x}) = \mathbf{I}_3$ and $\mathcal{B}(\mathbf{x}) = \mathbf{0}$ for the point \mathbf{x} such that $d^r(\mathbf{x}) < w^{\text{dist}}$ where w^{dist} is a user-specified distance. By using these adaptive metrics, we are able to model organs in complex CT volumes.

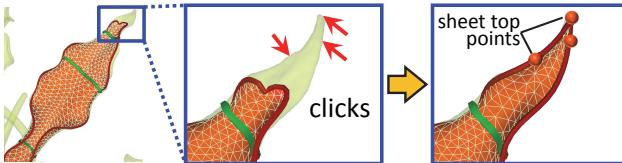


Figure 9: Additional sheet-top points specified to correct the boundary curve of the sheet at the top.

5 Numerical Scheme

In the previous section, we introduced three types of active curves (i.e., shaft axis, beam, and boundary curves), one active surface, and their energy functionals and corresponding metrics. To compute the active curves and surfaces interactively, we present an efficient numerical scheme consisting of shortest-path-based initialization and gradient descent optimization.

Our active curve and surface are approximated by polylines and triangle meshes, respectively. Minimizing the first terms in Eqs. (1) and (2) seeks to create non-uniform parameterizations in Ω_c and Ω_s but uniform vertex distributions on evolving curves and surfaces during optimizations. This is because minimizing these terms leads to bi-Laplacians in the corresponding Euler-Lagrange equations [Kobbelt et al. 1998], and the tangential component of the Laplacian (also bi-Laplacian) has mesh regularization effect [Wood et al. 2000; Ohtake and Belyaev 2001]. In addition to obtaining regular polylines and meshes, this effect works in our favor as a reparameterization to stabilize optimization steps. We employ fixed vertex positions and zero Laplacians at the boundary vertices during the both curve and surface optimizations. These Cauchy-type

boundary conditions provide interpolations with nicely smoothed connections to the fixed boundary.

5.1 Active Curve Fitting

Each of the three active curves is represented by a polyline $\mathbf{p} = (\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_N)$ that passes through the control points $\{q_k^\varphi\}$ where $\varphi = \{\text{shaft}, \text{beam}, \text{bound}\}$. The end points of $\{\mathbf{p}_i\}$ and $\{q_k^\varphi\}$ coincide. We discretize Eq. (1) with metrics (5-8) for a polyline \mathbf{p} by $E_c(\mathbf{p}) = \frac{1}{2} \sum_{i=2}^{N-1} |L_c(\mathbf{p}_i)|^2 +$

$$+ \alpha \sum_{i=1}^{N-1} \left| \left(\frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{h} \right)^T \frac{\mathcal{M}^\varphi(\mathbf{p}_{i+1}) + \mathcal{M}^\varphi(\mathbf{p}_i)}{2} \left(\frac{\mathbf{p}_{i+1} - \mathbf{p}_i}{h} \right) \right| \quad (10)$$

where $L_c(\mathbf{p}_i) = \frac{\mathbf{p}_{i+1} - 2\mathbf{p}_i + \mathbf{p}_{i-1}}{h^2}$ is a second-order central difference approximation of a Laplacian at \mathbf{p}_i with boundary conditions $L_c(\mathbf{p}_1) = L_c(\mathbf{p}_N) \equiv \mathbf{0}$ and the polyline vertices are resampled to have an equal interval h .

Polyline initialization. To obtain a polyline that minimizes Eq. (10), we adopt a shortest path algorithm to estimate a good initial approximation because otherwise, undesirable local minima produce erroneous fitting results. We construct a weighted graph over the foreground voxels V_f by representing each voxel as a graph node and each pair of neighboring voxels as an edge (Fig. 10a,b). We use 26 neighborhoods ($3 \times 3 \times 3$) for the edge. The edge length between two neighboring voxels, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^3$, is defined by using the second term of Eq. (10):

$$(\mathbf{a} - \mathbf{b})^T \left(\frac{\mathcal{M}^\varphi(\mathbf{a}) + \mathcal{M}^\varphi(\mathbf{b})}{2} \right) (\mathbf{a} - \mathbf{b}). \quad (11)$$

Unfortunately, the smoothness term $L_c(\cdot)$ cannot be incorporated because it requires three successive nodes. We then compute multiple shortest paths on this graph for all successive pairs of control points $(q_k^\varphi, q_{k+1}^\varphi)$ (Fig. 10c) by using the Dijkstra algorithm [Dijkstra 1959]. All paths are jointed sequentially and resampled with the interval h , resulting in an initial polyline, $\mathbf{p}^0 = (\mathbf{p}_1^0, \dots, \mathbf{p}_N^0)$.

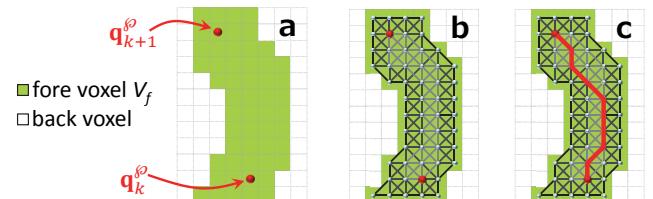


Figure 10: Polyline initialization. We construct a graph over V_f by connecting neighboring voxels (a,b); the edge length is defined by the Riemannian metric (11). A shortest path between successive control points is computed (c).

Gradient descent optimization. The initial polyline is a good approximation but usually contains noisy shapes. We update the inner vertices of the polyline $(\mathbf{p}_2, \mathbf{p}_3, \dots, \mathbf{p}_{N-1})$ by gradient descent iterations with the energy (10):

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k - \epsilon \frac{\partial E_c(\mathbf{p})}{\partial \mathbf{p}_i} \quad (12)$$

where the positions of two boundary vertices are unchanged during iterations (fixed boundary) and ϵ is a step size. The iteration stops when the mean change is smaller than a threshold. The derivative term of Eq. (12) is divided into two terms, $\frac{\partial E_c(\mathbf{p})}{\partial \mathbf{p}_i} = \frac{\partial E_c^1(\mathbf{p})}{\partial \mathbf{p}_i} + \frac{\partial E_c^2(\mathbf{p})}{\partial \mathbf{p}_i}$, corresponding to the first and second terms

in Eq. (10), respectively. The first term is calculated by the bi-Laplacian $\frac{\partial E_c^1(\mathbf{p})}{\partial \mathbf{p}_i} = L_c(L_c(\mathbf{p}_i)) = \frac{1}{h^2}(L_c(\mathbf{p}_{i+1}) - 2L_c(\mathbf{p}_i) + L_c(\mathbf{p}_{i-1}))$ with the boundary conditions $L_c(\mathbf{p}_1) = L_c(\mathbf{p}_N) \equiv 0$. We also estimate the second term by using the central difference on the volume grid, as it is difficult to derive a closed-form formula.

There are several minor implementation issues. Our metrics $\mathcal{M}^\varphi(\cdot)$ are based on the CT volume and then defined only at the centers of the grid voxels. The polyline vertices \mathbf{p}_i , in contrast, move freely in a 3D space. When computing $\mathcal{M}^\varphi(\mathbf{p}_i)$, we simply use the metric defined at the voxel nearest \mathbf{p}_i . This is for computational speed, and in our experience, this approximation works well. Also, all active curves should be placed inside the foreground V_f . All points of the polylines are checked after each iteration of Eq. (12), and if a point \mathbf{p}_i^{k+1} steps out of V_f , we pull it back as $\mathbf{p}_i^{k+1} = \mathbf{p}_i^k$. The beam curve is defined on a cross section, and thus a special treatment is required; when computing a shortest path for initialization, we consider only the voxels for which the distance to the cross section is less than twice the *voxel pitch* (actual edge length of a voxel). After each iteration in Eq. (12), the polyline vertices are projected onto the cross section.

5.2 Active Surface Fitting

A petal sheet is given by an active surface \mathbf{S} , which interpolates the curve network (Fig. 8d) and minimizes Eq. (2) with the matrix (9). We approximate \mathbf{S} with a triangle mesh (V, P) consisting of a set of vertices V and triangles P . Eq. (2) is discretized by

$$E_s(V, P) = \frac{1}{2} \sum_{\mathbf{v}_i \in V} |L_s(\mathbf{v}_i)|^2 + \beta \sum_{i, j, k \in P} |\bar{\mathcal{B}}(\mathbf{v}_j - \mathbf{v}_i) \times \bar{\mathcal{B}}(\mathbf{v}_k - \mathbf{v}_i)| \quad (13)$$

where \mathbf{v}_i is the i -th mesh vertex, $L_s(\mathbf{v}_i) = \mathbf{v}_i - \frac{1}{|N_i|} \sum_{j \in N_i} \mathbf{v}_j$ is a mesh Laplacian at \mathbf{v}_i with boundary conditions $L_s(\mathbf{v}_i) \equiv \mathbf{0}$ if $\mathbf{v}_i \in \partial \mathbf{S}$, N_i is a one-ring neighborhood of \mathbf{v}_i , and $\bar{\mathcal{B}}$ is evaluated at the triangle center $\bar{\mathcal{B}} = \mathcal{B}\left(\frac{\mathbf{v}_i + \mathbf{v}_j + \mathbf{v}_k}{3}\right)$.

Triangle mesh initialization. Similarly to the polyline initialization, we estimate an initial mesh by using the shortest path algorithm. Fig. 11 depicts our surface initialization procedure. Given a curve network of the sheet primitive (Fig. 11a), we divide it into multiple closed loops consisting of four curve segments: bottom, top, left, and right curves (Fig. 11b). For the top loop of the curve network, we split its boundary curve into three segments and obtain a loop formed by four curve segments (Fig. 11a). For each loop, the right and left curves are resampled with the same number of points. The bottom and top curves are resampled with the same predefined interval, and therefore may have different numbers of points, which is important because petal width is not uniform.

Assume that top and bottom curves consist of point sets indexed as $\{i\}_1^{N_t}$ and $\{j\}_1^{N_b}$, respectively. We build correspondences between them such that $j = \lfloor \frac{N_b-2}{N_t-2}(i-2) + 2 \rfloor$ if $N_t \geq N_b$; otherwise, $i = \lfloor \frac{N_t-2}{N_b-2}(j-2) + 2 \rfloor$. Then, multiple shortest paths between corresponding points are computed, as shown in Fig. 11c. The obtained vertical paths are resampled equally. Next, multiple horizontal shortest paths are computed such that each of them sequentially traverses corresponding points on the vertical paths (Fig. 11d). Finally, we equally resample each of the horizontal paths and fill all rectangular areas with triangles to obtain an initial mesh (V^0, P^0) (Fig. 11e). When computing the shortest path, a Euclidean metric $\mathcal{M}^\varphi = \mathbf{I}_3$ is applied for the graph edge length in Eq. (11) to avoid paths tracing petal's vascular bundles, which causes undesirable non-uniform sampling on the initial mesh.

Gradient descent optimization. Similarly to the curve fitting, we update the inner vertices of the initial mesh according to the gradient descent iterations

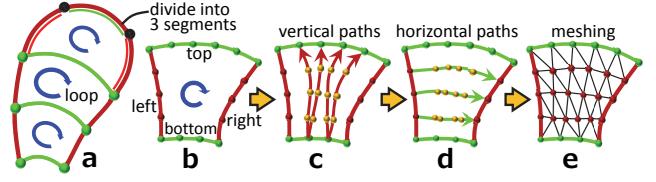


Figure 11: Surface initialization by multiple shortest paths.

gradient descent iterations

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k - \epsilon \frac{\partial E_s(V^k, P^k)}{\partial \mathbf{v}_i} \quad (14)$$

where $\mathbf{v}_i^k \in V^k$, the positions of boundary vertices are unchanged during iterations (fixed boundary), and ϵ is a step size. After each iteration, the mesh connectivity is updated so as to avoid skewed triangles by flipping edges if it provides the shorter edge. The iteration stops when the mean change is smaller than a threshold. The derivative term of Eq. (14) is divided into two terms, $\frac{\partial E_s(V^k, P^k)}{\partial \mathbf{v}_i} = \frac{\partial E_s^1}{\partial \mathbf{v}_i} + \frac{\partial E_s^2}{\partial \mathbf{v}_i}$, corresponding to the first and second terms of Eq. (13), respectively. The first term is calculated by the bi-Laplacian $\frac{\partial E_s^1}{\partial \mathbf{v}_i} = L_s(L_s(\mathbf{v}_i)) = L_s(\mathbf{v}_i) - \frac{1}{|N_i|} \sum_{j \in N_i} L_s(\mathbf{v}_j)$ with zero Laplacian at the boundary ($L_s(\mathbf{v}_i) \equiv \mathbf{0}$ if $\mathbf{v}_i \in \partial \mathbf{S}$). The second term is estimated by using the central difference on the volume grid. After fitting the surface, we compute front and back offsets, i.e., sheet thickness, at each vertex (Section 4.3).

6 Results and Discussion

The input flower volumes in this paper were taken by two CT devices: *Matsusada precision μRay8700* with a micro-focus X-ray tube (90kV/18W; max. voltage/power) and *Yxlon Y.CT Compact* (450kV) with a milli-focus X-ray tube (450kV/1.5kW; max. voltage/power). We performed modeling experiments on a machine with a 3.33 GHz Intel Core i7 processor and 24 GB RAM. Like the conventional active contours [Kass et al. 1988; Cohen and Cohen 1993], our active curve and surface models have several parameters. In all the examples in this paper, we used the following parameters: $\sigma = 1.5 \times l$ for Eq. (4) where l is a voxel pitch; $h = 6.0 \times l$ for curve resampling interval; $\epsilon = 0.005 \times h^4$ for Eq. (12); $\epsilon = 0.005$ for Eq. (14); $\alpha = \frac{0.01}{h^2}$ for Eq. (10); and $\beta = 0.01$ for Eq. (13).

Fig. 12 shows curve-fitting results for the shaft axis (a,b) and the boundary curve (c,d). The initial curves (a,c) estimated by the shortest path algorithms, based on Eq. (11), were placed close to desired positions but contained noisy shapes. They were then optimized by gradient descent iterations (12) (b,d). Fig. 13(a,b) shows our surface fitting results. An initial surface estimated by multiple shortest paths was placed very close to the target petal region (Fig. 13a). It was then optimized and fitted to the thin petal region by gradient descent iterations (14) (Fig. 13b). We stopped the gradient descent iterations when the mean change of vertices was less than $10^{-3} \times l$ where l is the voxel pitch. Because the estimated initial curves or surfaces were close to the desired locations, the iterations usually converged quickly. In our experience, they converged before 300 iterations for curves and before 1000 iterations for surfaces. In a $371 \times 393 \times 361$ volume, for example, optimizing a surface with 1794 vertices took less than 1.0 sec.

To confirm the effectiveness of our surface initialization with multiple shortest paths, we employed a thin plate surface [Kobbelt et al. 1998] computed from the curve network as the initial surface and applied gradient descent iterations (14) to it (Fig. 13c,d). As a result, a mesh locally fitted to a neighboring petal, and we could not obtain a desired petal mesh (see Fig. 13y,z). As in this example,

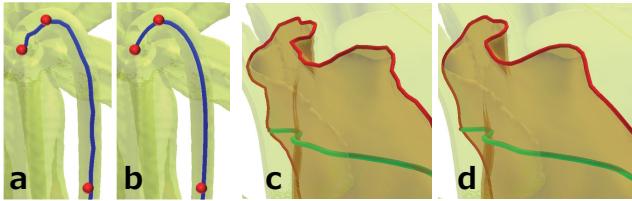


Figure 12: An axis curve of a receptacle (a,b) and a boundary curve of a petal (c,d). In each pair, the left pane indicates an initial estimate, and the right pane shows an optimized result.

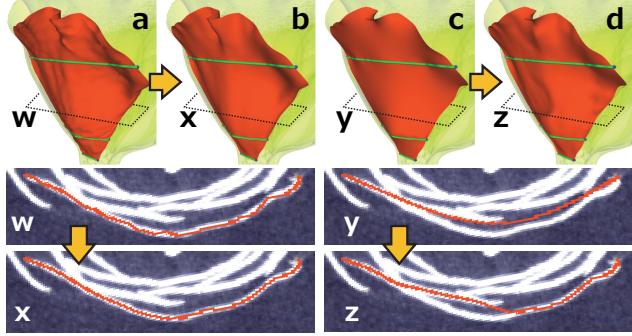


Figure 13: Petal surface fitting. We compute initial surfaces with our method (a) and with a thin plate surface (c), and iteratively optimize their shapes (b,d). The panes (w-z) show zoomed cross sections. See our supplemental material for high resolution images.

the initial surface mesh is especially important for CT-based flower modeling because a flower has multiple petals close to each other, and thus there are many undesired local minima around the target petal region. Our surface initialization technique avoids such local minima and provides aesthetic petal shapes in combination with the optimization (14).

Although the computational times for our primitive fitting depend on the resolution of the input volumes and the size of target organs, our technique works fast enough so as not to disturb real-time interaction. For example, the Eustoma model in Fig. 1 was constructed from an input volume with $371 \times 393 \times 361$ voxels, and fitting a stem/receptacle or pistil/stamen model took less than 0.5 sec. One of its petals, shown in Fig. 13b, has 1794 vertices, and computing its surface from three beam curves took 2.5 sec (including initialization, fitting, offsetting, and rasterization). We think that our implementation is fast enough for interactive flower modeling.

Figs. 1 and 14 show various flower models constructed with our technique by the authors. Although Y.CT Compact is designed for high-density materials (e.g., iron), our technique successfully reconstructs flowers with it (Fig. 14e,g). With our technique, flower organs are modeled by simple primitives, and thus it is easy to construct useful uv texture coordinates on them. We mapped textures obtained from photographs of real flowers onto the models in Figs. 1 and 14. Since these flower models were generated from CT volumes, their occluded internal organs were reconstructed correctly. Our technique successfully extracts petals contacting neighboring petals (Fig. 14a,h) and having wavy boundary shapes (Fig. 14d,g). All flower models and their input CT volumes are available on our web page [Ijiri 2014].

Table 1 provides statistical information for representative flower models. It took no more than 30 min interaction for modeling each flower model, except the Rosa model (Fig. 14h), which took about 2 hours (even though we omitted several difficult petals; see below). Table 1 also indicates that our technique requires, on average, about

1.7 min interaction time and 12.4 control points (mouse-clicks) to model a single petal. Modeling wavy petals via interactive software would require many more control points and more manipulation time; see our supplemental material for an example of interactive modeling software.

Model	Fig. 1	Fig. 14a	Fig. 14c	Fig. 14g	Fig. 14h
Total modeling time	30 min	27 min	15 min	14 min	106 min
Petals modeling time	21 min	20 min	7 min	8 min	100min
Petals (Beams/CPs)	20 (92/229)	15 (69/174)	6 (22/52)	6 (24/67)	47(229/647)
Voxel resolution (voxel pitch)	$371 \times 393 \times 361$ (0.17 mm)	$369 \times 365 \times 398$ (0.16 mm)	$321 \times 337 \times 487$ (0.14 mm)	$353 \times 389 \times 414$ (0.27 mm)	$482 \times 458 \times 243$ (0.23 mm)
Threshold / ω_{dist}	148 / 100	80 / 100	200 / 20	254 / 20	80 / 20

Table 1: Statistics of flower models. The third row shows the numbers of petals and sepals (beam curves/control points for modeling them). See our supplemental material for more detail.

Fig. 15 provides a comparison with three conventional segmentation methods: (b) iso-surfaces (thresholding with the marching cube [Lorensen and Cline 1987]), (c) graph cut [Boykov and Jolly 2001] with about 900 constraints on multiple cross sections, and (d) seeded region growing [Adams and Bischof 1994] with 20 seeds for each of five petals. The parameters and control points were specified carefully, but the results contain both topological and geometrical errors due to the absence of clear volume boundaries at regions of contacting petals. We believe that our primitive fitting technique is more suitable for extracting flower organs contacting one another without clear boundaries (Fig. 15a).

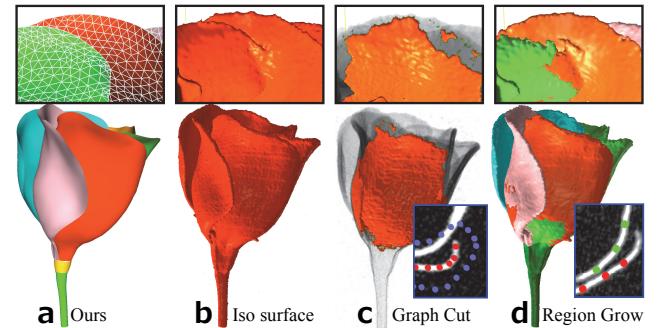


Figure 15: Comparison with conventional segmentation methods. Top row shows close-up views of a region with contacting petals. Red/blue points in pane (c) indicate fore/background voxels for the graph cut. Colored points in pane (d) are some of specified seeds.

Limitations and Future work. One limitation of our petal modeling technique is *self-contact*: it is difficult to deal with a petal having a boundary that contacts its own body surface. Such self-contact can be observed in the highly bent petals of the Rosa (Fig. 14h). Our mesh initialization method does not suppose such cases and therefore fails to obtain desired results (see Fig. 16a). Additionally, a self-contacting petal appears as a closed curve on the cross section as in Fig. 16b, and it is difficult to specify a beam curve with our current implementation. Providing a sophisticated method to handle such self-contact is a subject for our future work. Also, our technique requires flower CT resolution that is fine enough for the user to distinguish target organs on a cross section, because otherwise, it is difficult to specify valid control points (see Fig. 16 for fine (c) and failed (d) cases). For this reason, it was difficult to reconstruct pistils, stamens, and several internal petals of the Rosa (red and yellow arrows in Fig. 14h). Very high resolution X-ray CT systems might be useful for automatic flower reconstruction. On the other hand, they have their own limitations, such as imaging region size due to the fundamental mechanism of X-ray CT, and thus it is difficult to capture a whole flower with micro order details.

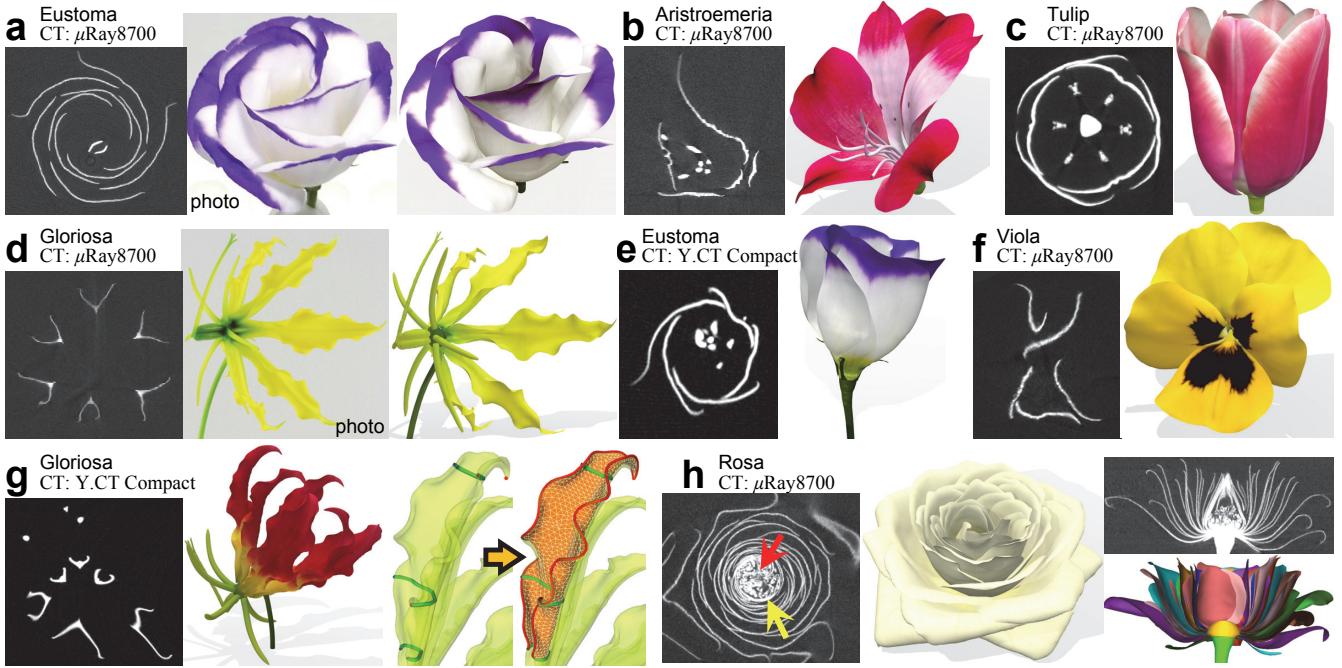


Figure 14: Our flower models with representative horizontal cross sections of their input CT volumes (a-h). The CT device used for modeling (i.e., μ Ray8700 or YCT Compact) is given in each pane. Panes (a) and (d) provide side-by-side comparisons with sample flower photographs. Panes (g) and (h) include a representative wavy petal and a cross section of the Rosa model, respectively.

Other future work includes presenting primitives for more complex flower organs (e.g., spiny petals or gamopetalous petals), developing automatic primitive-fitting techniques, extracting texture information from multiple photographs, and implementing implicit optimization schemes (e.g., [Eckstein et al. 2007]) to obtain faster convergence.

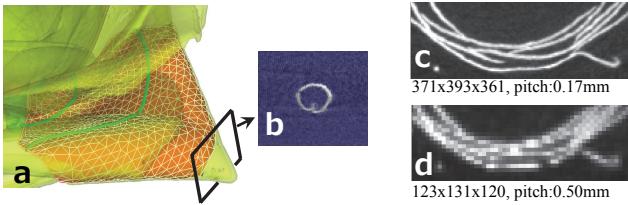


Figure 16: Limitations of our technique.

7 Conclusions

In this paper, we have proposed a novel flower modeling technique utilizing X-ray CT devices and real-world flowers. To our knowledge, this is the first reported attempt to use X-ray CT volumes for flower modeling. We assume that a flower mainly consists of shaft and sheet primitives, and we fit these primitives to the target volume regions semi-automatically. We introduced the novel active curve and surface formulations for the primitive fitting. We also developed an efficient numerical scheme to compute our active curves and surfaces based on the shortest path algorithm and gradient descent optimization. The resulting flower models demonstrate that our technique is promising for modeling realistic and complex flowers. We would like to emphasize that sheet-like and/or wavy shapes are commonly seen in the real world, such as in biological membranes and fabrics/clothing. We believe that our technique contributes to general curve and surface modeling from CT volumes.

Acknowledgments

We thank the anonymous reviewers for their valuable comments in improving this paper. This work was supported in part by the Grant-in-Aid for Scientific Research in Japan (25700011 and 24700182).

References

- ABEYSINGHE, S., AND JU, T. 2009. Interactive skeletonization of intensity volumes. *The Vis. Comput.* 25, 5-7, 627–635.
- ADAMS, R., AND BISCHOF, L. 1994. Seeded region growing. *IEEE Trans. PAMI.* 16, 6, 641–647.
- AMENTA, N., BERN, M., AND KAMVYSELIS, M. 1998. A new voronoi-based surface reconstruction algorithm. In *Proc. SIGGRAPH '98*, 415–421.
- ANASTACIO, F., PRUSINKIEWICZ, P., AND SOUSA, M. 2009. Sketch-based parameterization of L-systems using illustration-inspired construction lines and depth modulation. *Comput. Graph.* 33, 4, 440–451.
- BESSMELTSEV, M., WANG, C., SHEFFER, A., AND SINGH, K. 2012. Design-driven quadrangulation of closed 3D curves. *ACM Trans. Graph.* 31, 5.
- BLUM, H. 1967. A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, Ed. MIT Press, Cambridge, 362–380.
- BOYKOV, Y., AND JOLLY, M.-P. 2001. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. In *Proc. ICCV*, vol. 1, 105–112.
- BRADLEY, D., NOWROUZEZHARAI, D., AND BEARDSLEY, P. 2013. Image-based reconstruction and synthesis of dense foliage. *ACM Trans. Graph.* 32, 4, 74:1–74:10.

- CASELLES, V., KIMMEL, R., AND SAPIRO, G. 1997. Geodesic active contours. *IJCV* 22, 1, 61–79.
- COHEN, L., AND COHEN, I. 1993. Finite element methods for active contour models and balloons for 2D and 3D images. *IEEE Trans. PAMI* 15, 1131–1147.
- DIJKSTRA, E. 1959. A note on two problems in connexion with graphs. *Numer. Math.* 1, 269–271.
- ECKSTEIN, I., PONS, J.-P., TONG, Y., KUO, J., AND DESBRUN, M. 2007. Generalized surface flows for mesh processing. In *Proc. SGP*, 183–192.
- GLIMN-LACY, J., AND KAUFMAN, P. 2006. *Botany Illustrated: Introduction to Plants, Major Groups, Flowering Plant Families*. Springer.
- GREGSON, J., KRIMERMAN, M., HULLIN, M., AND HEIDRICH, W. 2012. Stochastic tomography and its applications in 3D imaging of mixing fluids. *ACM Trans. Graph.* 31, 4.
- HUANG, H., WU, S., COHEN-OR, D., GONG, M., ZHANG, H., LI, G., AND CHEN, B. 2013. L1-medial skeleton of point cloud. *ACM Trans. Graph.* 32, 65:1–65:8.
- IJIRI, T., OWADA, S., OKABE, M., AND IGARASHI, T. 2005. Floral diagrams and inflorescences: Interactive flower modeling using botanical structural constraints. *ACM Trans. Graph.* 24, 3, 720–726.
- IJIRI, T., YOSHIZAWA, S., SATO, Y., ITO, M., AND YOKOTA, H. 2013. Bilateral Hermite radial basis functions for contour-based volume segmentation. *Comput. Graph. Forum* 32, 2, 123–132.
- IJIRI, T. 2014. Flower CT volume and surface model library. www.riken.jp/brict/Ijiri/flowerctlib/.
- KASS, M., WITKIN, A., AND TERZOPoulos, D. 1988. Snakes: Active contour models. *IJCV* 1, 4, 321–331.
- KAZHDAN, M., AND HOPPE, H. 2013. Screened poisson surface reconstruction. *ACM Trans. Graph.* 32, 3, 29:1–29:13.
- KOBBELT, L., CAMPAGNA, S., VORSATZ, J., AND SEIDEL, H.-P. 1998. Interactive multi-resolution modeling on arbitrary meshes. In *Proc. ACM SIGGRAPH'98*, 105–114.
- LI, Y., FAN, X., MITRA, N., CHAMOVITZ, D., COHEN-OR, D., AND CHEN, B. 2013. Analyzing growing plants from 4D point cloud data. *ACM Trans. Graph.* 32, 6, 157:1–157:10.
- LINDENMAYER, A. 1968. Mathematical models for cellular interaction in development: Parts I and II. *J. Theor. Bio.* 18.
- LINTERMANN, B., AND DEUSSEN, O. 1999. Interactive modeling of plants. *IEEE Comput. Graph Appl.* 19, 1, 56–65.
- LONGAY, S., RUNIONS, A., BOUDON, F., AND PRUSINKIEWICZ, P. 2012. Treesketch: Interactive procedural modeling of trees on a tablet. In *Proc. SBIM*, 107–120.
- LORENSEN, W., AND CLINE, H. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph.* 21, 4, 163–169.
- MORETON, H. P., AND SÉQUIN, C. H. 1992. Functional optimization for fair surface design. *Comput. Graph.* 26, 2, 167–176.
- MĚCH, R., AND PRUSINKIEWICZ, P. 1996. Visual models of plants interacting with their environment. In *Proc. SIGGRAPH '96*, 397–410.
- NEALEN, A., IGARASHI, T., SORKINE, O., AND ALEXA, M. 2007. Fibermesh: Designing freeform surfaces with 3D curves. *ACM Trans. Graph.* 26, 3, 41:1–41:10.
- OHTAKE, Y., AND BELYAEV, A. 2001. Mesh optimization for polygonized isosurfaces. *Comput. Graph. Forum* 20, 3, 368–376.
- OKABE, M., OWADA, S., AND IGARASHI, T. 2005. Interactive design of botanical trees using freehand sketches and example-based editing. *Comput. Graph. Forum* 24, 3, 487–496.
- OSHER, S., AND SETHIAN, J. 1988. Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* 79, 1, 12–49.
- PALUBICKI, W., HOREL, K., LONGAY, S., RUNIONS, A., LANE, B., MĚCH, R., AND PRUSINKIEWICZ, P. 2009. Self-organizing tree models for image synthesis. *ACM Trans. Graph.* 28, 3, 58:1–58:10.
- PRUSINKIEWICZ, P., AND LINDENMAYER, A. 1990. *The Algorithmic Beauty of Plants*. Springer-Verlag New York, Inc.
- PRUSINKIEWICZ, P., JAMES, M., AND MĚCH, R. 1994. Synthetic topiary. In *Proc. SIGGRAPH '94*, 351–358.
- PRUSINKIEWICZ, P., MÜNDERMANN, L., KARWOWSKI, R., AND LANE, B. 2001. The use of positional information in the modeling of plants. In *Proc. SIGGRAPH '01*, 289–300.
- QUAN, L., TAN, P., ZENG, G., YUAN, L., WANG, J., AND KANG, S. 2006. Image-based plant modeling. *ACM Trans. Graph.* 25, 3, 599–604.
- RECHE-MARTINEZ, A., MARTIN, I., AND DRETTAKIS, G. 2004. Volumetric reconstruction and interactive rendering of trees from photographs. *ACM Trans. Graph.* 23, 3, 720–727.
- SAUVOLA, J., AND PIETIKÄINEN, M. 2000. Adaptive document image binarization. *Pattern Recognit.* 33, 225–236.
- SETHIAN, J. 1999. *Level Set Methods and Fast Marching Methods*. Cambridge University Press.
- SHARF, A., LEWINER, T., SHAMIR, A., KOBBELT, L., AND COHEN-OR, D. 2006. Competing fronts for coarse-to-fine surface reconstruction. *Comput. Graph. Forum* 25, 3, 389–398.
- SOCHEN, N., KIMMEL, R., AND MALLADI, R. 1998. A general framework for low level vision. *IEEE Trans. Image Proc.* 7, 3, 310–318.
- STUPPY, W., MAISANO, J., COLBERT, M., RUDALL, P., AND ROWE, T. 2003. Three-dimensional analysis of plant structure using high-resolution X-ray computed tomography. *Trends Plant Sci.* 8, 1, 2–6.
- SÜSSMUTH, J., AND GREINER, G. 2007. Ridge based curve and surface reconstruction. In *Proc. SGP*, 243–251.
- TAN, P., FANG, T., XIAO, J., ZHAO, P., AND QUAN, L. 2008. Single image tree modeling. *ACM Trans. Graph.* 27, 5, 108:1–108:7.
- TRIFONOV, B., BRADLEY, D., AND HEIDRICH, W. 2006. Tomographic reconstruction of transparent objects. In *Proc. EGSR*, 51–60.
- WOOD, Z., SCHRÖDER, P., BREEN, D., AND DESBRUN, M. 2000. Semi-regular mesh extraction from volumes. In *Proc. IEEE Vis*, 275–282.
- ZHANG, C., YE, M., FU, B., AND YANG, R. 2014. Data-driven flower petal modeling with botany priors. In *Proc. CVPR (to appear)*.
- ZHAO, S., JAKOB, W., MARSCHNER, S., AND BALA, K. 2011. Building volumetric appearance models of fabric using micro CT imaging. *ACM Trans. Graph.* 30, 4, 44:1–44:10.