

Research Paper:

Leaf Reconstruction Based on Gaussian Mixture Model from Point Clouds of Leaf Boundaries and Veins

Yukie Nagai[†] and Hikaru Tanaya

Graduate School of Systems Design, Tokyo Metropolitan University

6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

[†]Corresponding author, E-mail: y-nagai@tmu.ac.jp

[Received June 30, 2023; accepted November 7, 2023]

Three-dimensional (3D) models of leaves are expected to contribute to a wide range of applications, including the study of plant morphology and leaf design. Leaf boundaries and veins are key factors in determining leaf shape in both botany and design. This motivated us to design a leaf-shape generator that uses leaf boundaries and veins. We propose an algorithm to reconstruct leaf geometry as a surface mesh generated from point clouds of leaf boundaries and veins. First, it determines the interior region of the leaf using the multi-level partition of unity implicit approach. Then, based on the Gaussian mixture model, it expresses the 3D shape of the leaf, where the values vary depending on the distances from the leaf boundary to veins. The use of differentiable functions for leaf shapes realizes smooth underlying surfaces and enables various shape analyses using differential operations.

Keywords: leaves, point clouds, surface mesh, Gaussian mixture model, implicit functions

1. Introduction

Plants exhibit a wide variety of shapes that reflect their habitat environments, such as temperate and tropical zones and deserts, for better adaptation resulting from long-term evolution. This motivates research on plant geometry to determine vegetation evolution. Research on the evolution of plant shapes is based on many research fields, including morphology, plant physiology, comparative morphology, comparative biology, taxonomy, and molecular biology. Furthermore, to these traditional biological approaches, computational analyses of plant shapes can reveal plant evolutionary processes.

Recently, research on plant physiology determined that various plant hormones determine plant geometries. For instance, the hormone auxin induces growth and the circling movement of plants when they are exposed to light. Another hormone, gibberellin, contributes to the thickness of plants. The elongational growth of plant cells depends on the extensibility of the cell walls. A cell wall consists of bundles of cellulose fibers that extend when the bundles are loosened. Gibberellin determines the direction in

which the cellulose fibers are loosened. Elongation and winding are movements in space; thus, three-dimensional (3D) analysis is desirable. However, the representation analysis of plants in three dimensions is complicated, and it makes plant shape analysis a two-dimensional (2D) process, for example, observations using microscopes.

This motivated us to propose an intuitive shape reconstruction algorithm for leaves that uses simple inputs essential for the determination of leaf shapes to enable users unfamiliar with shape modeling to easily generate 3D leaf surfaces. We expect that this simple setting will lead to its use with many applications, for instance, growth expectations of leaves based on actual geometry data of boundaries and veins obtained using 3D scanning technologies, such as optical scanning and X-ray computed tomography (CT) scanning, of young leaves. To apply this algorithm to various shape analyses, for many of which differential values of shapes are adopted, shape representation using continuous and differentiable expressions is required. Aimed at finding a solution for this purpose, we propose surface reconstruction specialized to leaves. In Section 2, we discuss existing surface reconstruction algorithms and leaf modeling methods. In Section 3, the details of our algorithm are explained. The experimental results are presented in Section 4. Section 5 presents a summary and suggests future work.

2. Related Work

Surface reconstruction generates a surface mesh from a set of points (point cloud) sampled from the surface of an object by determining the location and connectivity of the points to retrieve the geometry and topology of the scanned object [1–4]. Point clouds are generally generated from the entire surface of objects using optical scanners [5]. This technology is useful for many industrial applications, and high demand has created a lot of research in this field over the past 30 years. The algorithms are classified into two types of approaches: explicit algorithms that utilize Delaunay triangulation or its dual structure, Voronoi tessellation, of the input point cloud; and implicit algorithms that generate scalar fields and extract isosurfaces that approximate a surface from which the input points are sampled.



Explicit algorithms were developed from the cocone [6] and crust [7] algorithms. While they mathematically guarantee the accuracy of a generated surface from object's surface under conditions of the sampling density, they are not very robust against noise and lack of sampling due to the Delaunay triangulation and Voronoi tessellation, which they utilize to determine the topology of the surface. Therefore, variations [8–10] have been developed to overcome this issue. Other algorithms in this category include alpha shape [11, 12], which can generate meshes with desired levels of detail.

Implicit surface reconstruction was established by [2]. The advantages of implicit methods are that they can produce smooth surfaces by approximating or interpolating input points to generate scalar fields. To generate suitable scalar fields, many diverse approaches have been proposed, including the use of radial basis functions and their localization [13, 14], moving least squares [15, 16], wavelet [17], and point set surfaces [18, 19]. Implicit algorithms are essentially good at handling a lack of sampling, and graph-cut was introduced to further enhance this advantage [20]. Multi-level partition of unity implicits (MPU) approaches realize robust and fast computation [21, 22]. Other well-known implicit algorithms include Poisson equation-based algorithms [23, 24], which produce accurate and smooth surfaces.

Recently, the machine learning approach has been enthusiastically developed. Point2Mesh robustly generates surfaces even from point clouds that lack sampling [25]. DeepSDF [26] generates signed distance functions for various categories of objects using smaller memory consumption than conventional surface reconstruction algorithms with the help of the autodecoder learning method. To overcome the limitation of DeepSDF, a global method, DeepLS [27] was proposed to localize the generation of signed distance fields, which achieved better results. OccNet [28] generates implicit reconstructions of 3D scenes of detailed objects by introducing implicit occupancy decoders to convolutional encoders.

As discussed, surface reconstruction is a well-studied topic, and comprehensive information about it can be found in Berger et al.'s extensive summary of surface reconstruction algorithms [29].

However, while there are fewer plant modeling algorithms, the data structures of their inputs and outputs are more diverse than those of typical surface reconstruction algorithms. Most research has attempted to generate 3D leaf models using 2D images as the input [30, 31]. Other research has used point clouds sampled from leaf surfaces, which are then reconstructed as meshes to represent the leaves [32]. However, the bottlenecks of these inputs are that they are insufficient to represent the target 3D shapes, or that obtaining a point cloud of the entire leaf requires specialized skills and special devices for optical scanning. Conversely, the point clouds of boundaries and veins are 3D data, more intuitive, and more readily available than surface data sets. Point clouds can be generated using mathematical equations, CAD software, and image sampling, or acquired using 3D scanning, such as X-ray CT.

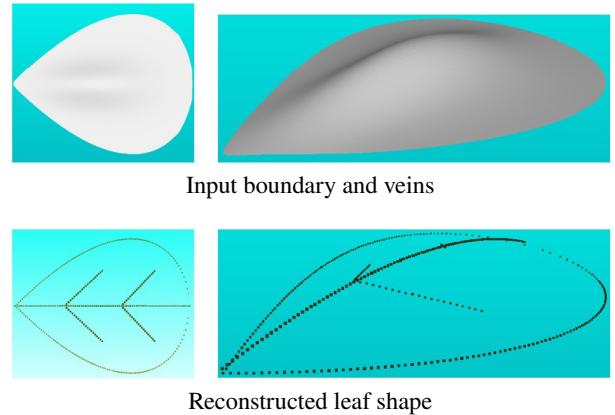


Fig. 1. Examples of the input point cloud (top) and output surface (bottom) of the proposed surface reconstruction algorithm.

For these reasons, it would be advantageous to easily generate 3D leaf models from the point clouds of the essential leaf structures that can be used for various applications, including plant growth prediction when no grown leaves are available.

3. Algorithm

3.1. Outline

We propose an algorithm for modeling 3D leaf shapes that uses only point clouds of leaf veins and boundaries, both of which play a key role in the figuration of entire leaves. The input boundary points are supposed to be a planar convex curve. We assume that the vein point clouds are divided by veins. This is essentially possible when each vein is generated from a mathematical expression. In other cases, a segmentation algorithm or manual segmentation using software such as MeshLab [33] can be used. The output is a surface mesh that follows the geometry suggested by the veins, which is trimmed at the boundary and does not spread infinitely (**Fig. 1**).

This process is a variation of surface reconstruction; however, it differs from conventional surface reconstruction methods in that, generally, surface reconstruction expects input points sampled from the entire surface of an object, but our proposed method does not. The proposed method expresses leaf geometry using a Gaussian mixture model (GMM) to realize a smooth surface applicable to many shape analyses that require differentials of the subject. The flow of the algorithm, as follows, is illustrated in **Fig. 2**:

- Step 1. Project the boundary points onto a plane.
- Step 2. Detect the leaf region as $\{x | g(x) \geq 0\}$ by generating an implicit function $g(x)$.
- Step 3. Estimate the probability $p(x)$.
- Step 4. Generate a GMM whose probability conforms to $p(x)$.

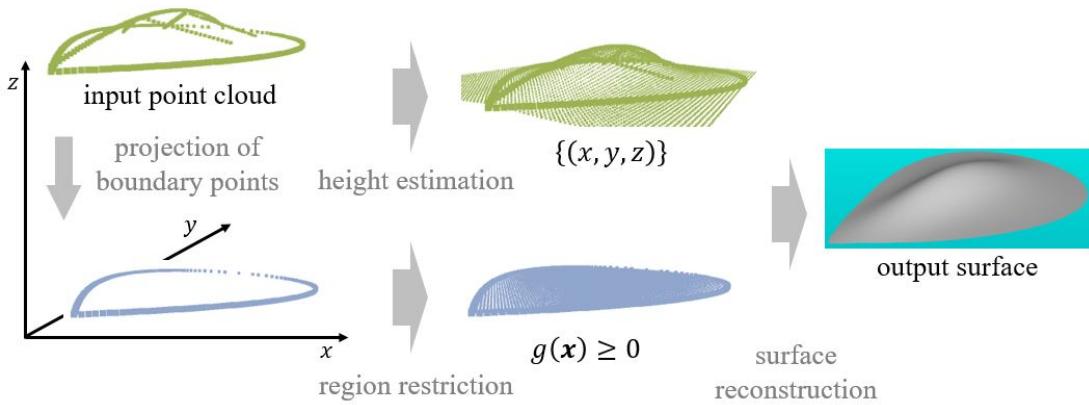


Fig. 2. Algorithm flow.

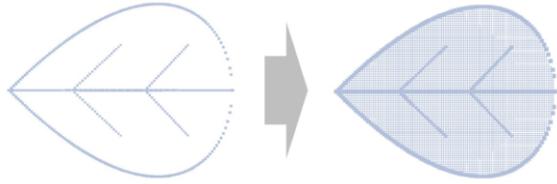


Fig. 3. Interior restriction by MPU [21]. Left: input point clouds, right: the interior region (hatched area).

Step 5. Reconstruct the surface for points sampled from the graph of the GMM.

3.2. Interior Restriction

The first step of the leaf reconstruction is the restriction of the interior region (**Fig. 3**). For this purpose, the input boundary (planar curve) is firstly projected onto a plane. The inner region on the plane is specified by a 2D version of the implicit surface reconstruction algorithm, which generates a scalar field with different signs in the inner and outer regions. We adopted the MPU [21] approach because it is a local approximation algorithm and expresses small features well. This enables the retrieval of small features, such as the tips of narrow leaves, as seen, for example, on bamboo leaves. A scalar function $g(x, y)$, generated by MPU and defined on the projection plane, returns a positive, zero, and negative value for the interior, boundary, and exterior of the leaf, respectively. The function is formulated as follows:

$$g(\mathbf{x}) = \frac{\sum w_i(\mathbf{x}) f_i(\mathbf{x})}{\sum w_i(\mathbf{x})}, \quad \dots \quad (1)$$

where $w_i(\mathbf{x})$ is a weighting function and $f_i(\mathbf{x})$ is a locally supported function that approximates the boundary shape using a zero-level set. In this case, it is represented by the linear function $f_i(\mathbf{x}) = c_{i1}x + c_{i2}y + c_{i3}$ (c_{i*} is a coefficient). The support of each local approximation is dynamically determined to maximize the size r with an approximation error less than the tolerance value. As the weighting func-

tion, for a support with a center \mathbf{c}_i and a radius r_i , we adopted a B-spline function $w_i(\mathbf{x}) = b\{(3|\mathbf{x} - \mathbf{c}_i|)/2r_i\}$, defined as follows:

$$b(t) = \begin{cases} 0, & 1 \leq t; \\ \frac{9}{8}(t-1)^2, & \frac{1}{3} \leq t < 1; \\ -\frac{9}{4}t^2 + \frac{3}{4}, & 0 \leq t < \frac{1}{3}. \end{cases} \quad (2)$$

3.3. Leaf Height Estimation

The surfaces of most leaves are highest on their veins, and the surface smoothly lowers toward the leaf boundaries. We expressed this tendency using a GMM that produces height values. The GMM can generate a smoothly curved surface from estimated heights, which are later given by linear forms. Other surface reconstructions fit more strictly to given points and therefore may generate more flat surfaces.

The complete algorithm for this is presented below and shown in **Fig. 4**. Precise descriptions of each operation are presented in the following paragraphs.

Step 1. Project all the boundary points onto a plane P .

Step 2. Compute the height from P at all the vein points.

In Step 1, the plane P : $ax + by + cz + d = 0$ for the boundary points to be projected is constructed as a plane whose normal is the normalized eigenvector of the minimum eigenvalue of the principal component analysis for the boundary points and passes through the average point of the input points.

Generally, P generated in Step 1 is not parallel to the $x-y$ plane of the Cartesian coordinate system. The height h_v at a vein point (x_i, y_i, z_i) from P is simply computed as the distance between the point and P as follows:

$$h_v = \frac{|ax_i + by_i + cz_i + d|}{\sqrt{a^2 + b^2 + c^2}}. \quad \dots \quad (3)$$

The height at a general point on a leaf is a little closer to the boundary, vanishes at the boundary, and increases to-

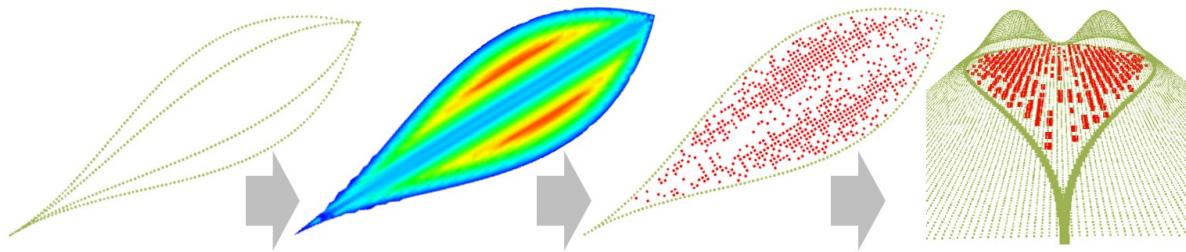


Fig. 4. Generation of the GMM. From left to right, input points, color map of $p(\mathbf{x})$, auxiliary points (red), and points sampled from the GMM (green).

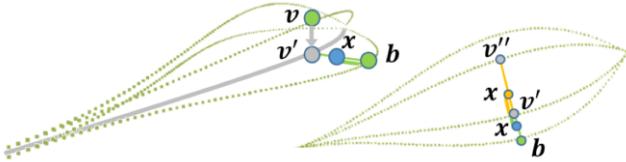


Fig. 5. Leaf probability estimation. Left: projection of a vein point v to the projection plane. Right: two cases (distances required for the calculation are indicated in green and orange) of estimated point locations (top view).

ward the vein. This observation suggests that determining the height depends on the distance from the input points. The probability, calculated by Eqs. (4) and (5), changes its value linearly; therefore, this is not appropriate for leaf shapes. However, the GMM generates smoothly curved surfaces based on these values. Let \mathbf{b} and v be the closest boundary and vein points, respectively, to the query point \mathbf{x} , and a point v' be the projected point of v on the plane (see the left image in **Fig. 5**). The probability $p(\mathbf{x})$ at a point \mathbf{x} between the boundary and a vein is determined by the height h_v at v and the ratio of the distance from the boundary to \mathbf{x} to the distance from the boundary to the vein $|\mathbf{b} - v'|$ is shown in the following equation:

$$p(\mathbf{x}) = h_v \frac{|\mathbf{b} - \mathbf{x}|}{|\mathbf{b} - v'|}. \quad \dots \quad (4)$$

When there are multiple closest points \mathbf{b} and v , the average of $|\mathbf{b} - v'|$ for all combinations of \mathbf{b} and v is used as a denominator, and the average of their heights is used as h_v . This is illustrated in the left image and in the bottom part of the right image in **Fig. 5**, where the distances are indicated by the green segments.

The case, where the point's two closest height constraints are not a combination of a vein and boundary, but rather two veins as parallel veins, is shown in the center of the right image in **Fig. 5**, and the probability is as follows:

$$p(\mathbf{x}) = h_v \left(1 - \frac{|\mathbf{v}' - \mathbf{x}| |\mathbf{v}' - \mathbf{v}''|}{2} \right), \quad \dots \quad (5)$$

where v'' is the second closest point on a vein. This equation gives a point within two veins a probability that depends on the height of the closer vein and decreases away from the veins. See the center of the right image in **Fig. 5**,



Fig. 6. Concept of the GMM.

where the distances for this calculation are indicated by the orange segments. An example of a color map of the probability is shown in the second left image in **Fig. 4**. The value is high in the red region and low in the blue region.

3.4. GMM Generation

Modeling: For better surface reconstruction results, sample points should be used that smoothly transit the height from the veins to the boundary and are evenly sampled. Therefore, we generated a function, GMM, and sampled its graph as shown in the right image in **Fig. 4**. It represents the existence probability of the given points and shows a large value (high probability) where it has a high density of given points. Thus, we considered the probability $p(\mathbf{x})$ determined in the previous step as the guide for the existence probability and generated points in the distribution conformal to $p(\mathbf{x})$. Examples of the probability and obtained samples are shown in the second left and second right images of **Fig. 4**. The red-coded regions in the color map have more points than the blue-coded areas, which have fewer samples.

Mathematical expression: The GMM integrates Gaussian functions to represent multiple peaks (**Fig. 6**). A Gaussian function $N(\mathbf{x}|\boldsymbol{\mu}, \Sigma)$ for a 2D data set $\mathbf{x} = (x, y)$ is

$$N(\mathbf{x}|\boldsymbol{\mu}, \Sigma) = \frac{1}{2\pi} \frac{1}{|\Sigma|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^T \Sigma^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\} \quad (6)$$

for a mean vector $\boldsymbol{\mu}$ and a covariance matrix Σ

$$\begin{cases} \boldsymbol{\mu} = (\mu_x, \mu_y)^T, \\ \Sigma = \begin{pmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{pmatrix}. \end{cases} \quad \dots \quad (7)$$

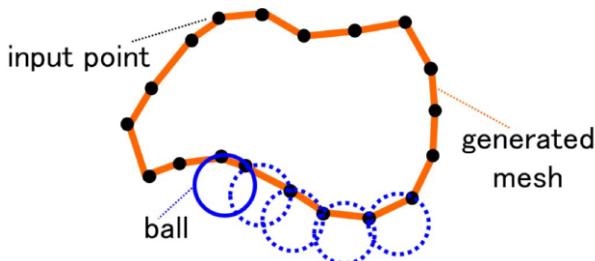


Fig. 7. Schematic image of ball pivoting for 2D input.

In the above equation, $|\Sigma|$ means the determinant of Σ . A GMM is defined by K Gaussian functions, each of which is $N(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k)$, with its mean vector $\boldsymbol{\mu}_k$ and covariance matrix Σ_k ($k = 1, \dots, K$) described as follows:

$$z = \sum_{k=1}^K \pi_k N(\mathbf{x}|\boldsymbol{\mu}_k, \Sigma_k) \quad \dots \quad (8)$$

The weight π_k satisfies the following restriction:

$$\sum_{k=1}^K \pi_k = 1, \quad 0 \leq \pi_k \leq 1. \quad \dots \quad (9)$$

To generate a GMM, we determined the values π_k , $\boldsymbol{\mu}_k$, and Σ_k from the points generated in the previous step so that z conforms to $p(\mathbf{x})$. These values can be computed as the maximum likelihood solutions by an EM algorithm [34].

The value of the GMM is evaluated at each grid point on the x - y plane. An example of the points sampled in this way is the green points in the rightmost image in **Fig. 4**, which are overlaid on the samples from Step 3 (red points) and the input boundary.

3.5. Surface Reconstruction

To extract the surface, first, the points $\{(x, y, z)\}$, where z is given by Eq. (8) satisfying $g(\mathbf{x}) \geq 0$ for $\mathbf{x} = (x, y)$, should be extracted from the point set sampled from the GMM graph. Second, a surface mesh is reconstructed using ball pivoting [35]. This is an explicit surface reconstruction algorithm and does not generate a surface outside the boundary. A schematic image of ball pivoting is shown in **Fig. 7**.

4. Results

The experiments were conducted on a machine with an Intel Core i7-8700, 3.20 GHz CPU, and 32 GB of RAM. We applied the proposed algorithm to two models: one with nested veins and one with parallel veins. Many types of software are applicable to generating boundary points; therefore, we simply sampled them from graphs of polynomial equations. The value of K was determined based on observations.

The first example (nested veins) comprises a total of

383 vein and boundary points. In this case, we set K to 3. **Fig. 8** indicates the input point cloud and generated surface. The equation of the boundary is $y = x^2(1 - x)$, as shown in the bottom left in **Fig. 8** and its reflection. The veins in (a) and (d) were also given by sampling linear equations. As in (c), the generated surface fits to the boundary well, and its height conforms to the height suggested by the center vein.

Figure 9 shows parallel veins with 402 points. The value of K was set to 2. Half of the boundary was defined by $y = x\sqrt{1 - x}$, and its graph is shown in the bottom left in **Fig. 9**. The other images in this figure show that the generated surfaces have pointed tips, as in the input boundary, and it has smooth slopes with peaks on the suggested veins.

We also compared our result with a common surface reconstruction algorithm, the screened Poisson surface reconstruction [24], for nested veins. The results are shown in **Fig. 10**. While the screened Poisson surface reconstruction (**Fig. 10(c)**) generated a wide surface outside the boundary due to the unexpected character of the input point set, the proposed algorithm gave a surface (**Fig. 10(b)**) that was correctly truncated at the boundary.

We quantitatively evaluated the deviation from the input points and the generated leaves. The distances from the input points are shown in **Table 1**. The entire sizes of the leaves are 1 in their longest direction (see boundary shapes in **Figs. 8** and **9**). This shows that in both cases, nested veins and parallel veins, the generated surfaces conform to the input.

5. Conclusion

We proposed an algorithm to generate leaf surfaces via a GMM from point clouds of leaf boundaries on a plane and of leaf veins. This strategy can successfully represent smooth differentiable surfaces, which are highest on the veins and conformal to the intended boundary shape. This differentiable surface is applicable to geometry analysis, for instance, using differentials as curvatures.

For more flexibility of design, the algorithm will be improved to allow for non-planar boundaries and concave curves. This will widen the applicable curves, for instance, ones generated by sketch-based curve modeling tools, and allow evaluation by comparison to 3D-scanned data of real leaves. With a 3D curve as input, one solution is to project the curve onto a plane and apply a deformation technique to the surface generated from the planar boundary. Currently, the value of K is determined based on observations; therefore, automatic determination of this value will improve the utility of the proposed algorithm. Changing the model from a GMM to other smooth functions will also expand the range of expressible leaf shapes. To account for vein crease lines on the surface of real leaves, deformation as used in [31] for the same purpose can be used after generating a mesh.

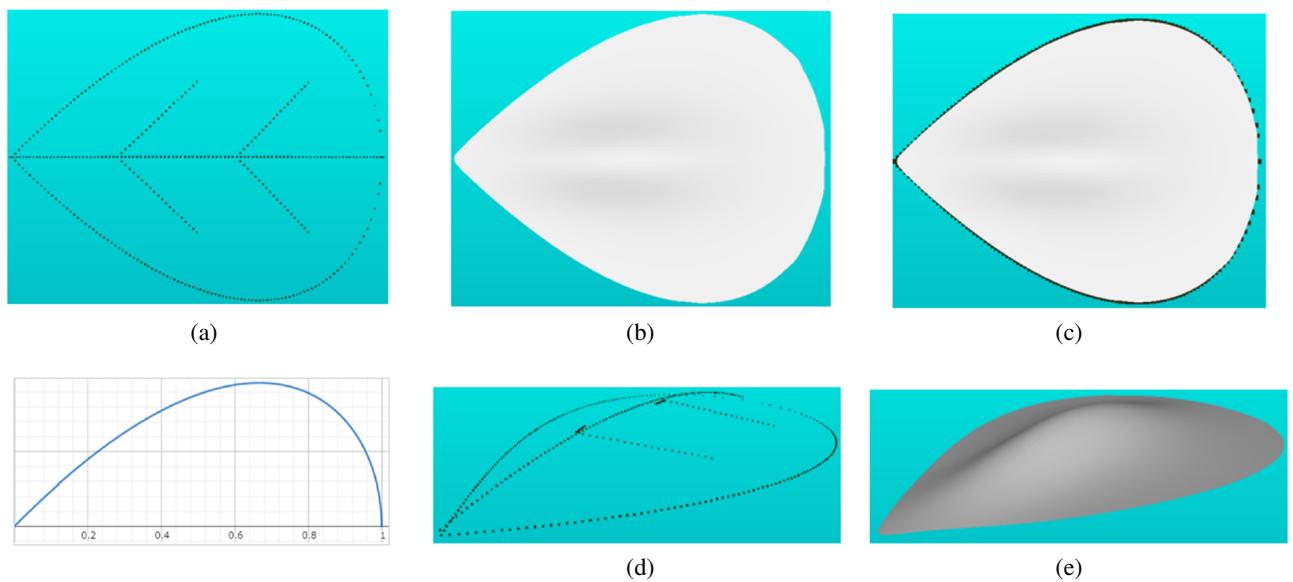


Fig. 8. Result for nested veins. (a) Input, (b) generated surface, (c) surface in (b) overlaid on the input in (a), and (d) and (e) other views of (a) and (b).

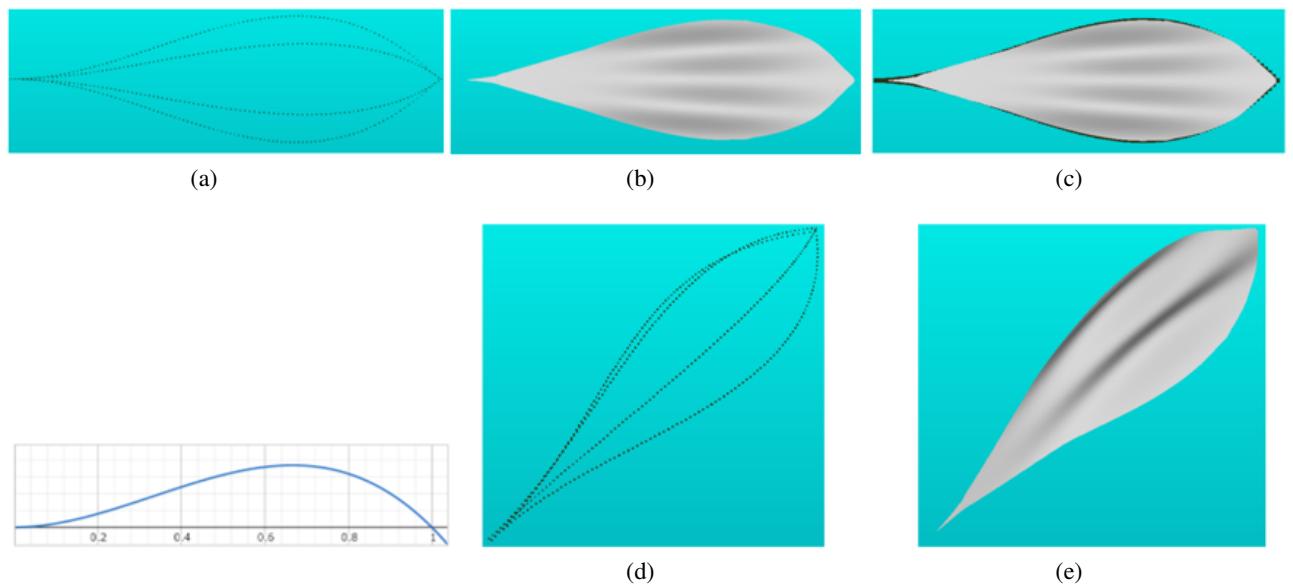


Fig. 9. Result for parallel veins. (a) Input, (b) generated surface, (c) surface in (b) overlaid on the input in (a), and (d) and (e) other views of (a) and (b).

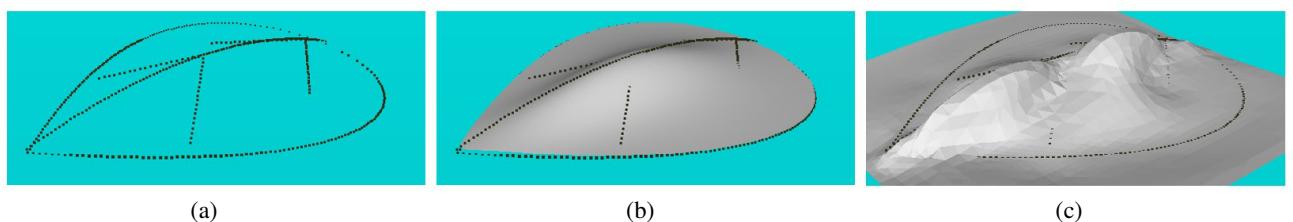


Fig. 10. Comparison with a general surface reconstruction for nested veins. (a) Input, (b) our result, and (c) result from screened Poisson surface reconstruction.

Table 1. Distances from the input points to the generated surface.

| Type | From boundary | From veins |
|----------|---------------------------|---------------------------|
| Nested | min: 2.80×10^3 | min: 1.00×10^6 |
| | max: 8.77×10^3 | max: 28.98×10^3 |
| | mean: 4.25×10^3 | mean: 9.77×10^3 |
| Parallel | min: 2.75×10^3 | min: 2.40×10^5 |
| | max: 31.98×10^3 | max: 31.43×10^3 |
| | mean: 17.37×10^3 | mean: 11.41×10^3 |

References:

- [1] J.-D. Boissonnat, “Geometric structures for three-dimensional shape representation,” ACM Trans. Graph., Vol.3, No.4, pp. 266-286, 1984. <https://doi.org/10.1145/357346.357349>
- [2] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle, “Surface reconstruction from unorganized points,” ACM SIGGRAPH Comput. Graph., Vol.26, No.2, pp. 71-78, 1992. <https://doi.org/10.1145/142920.134011>
- [3] B. Curless and M. Levoy, “A volumetric method for building complex models from range images,” Proc. 23rd Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’96), pp. 303-312, 1996. <https://doi.org/10.1145/237170.237269>
- [4] G. Turk and M. Levoy, “Zippered polygon meshes from range images,” Proc. 21st Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’94), pp. 311-318, 1994. <https://doi.org/10.1145/192161.192241>
- [5] H.-C. Nguyen and B.-R. Lee, “3D model reconstruction system development based on laser-vision technology,” Int. J. Automation Technol., Vol.10, No.5, pp. 813-820, 2016. <https://doi.org/10.20965/ijat.2016.p0813>
- [6] T. K. Dey and J. Giesen, “Detecting undersampling in surface reconstruction,” Proc. 17th Annu. Symp. Comput. Geom. (SCG’01), pp. 257-263, 2001. <https://doi.org/10.1145/378583.378682>
- [7] N. Amenta, M. Bern, and M. Kamvysselis, “A new Voronoi-based surface reconstruction algorithm,” Proc. 25th Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’98), pp. 415-421, 1998. <https://doi.org/10.1145/280814.280947>
- [8] T. K. Dey and S. Goswami, “Tight cocone: A water-tight surface reconstructor,” Proc. 8th ACM Symp. Solid Model. Appl. (SM’03), pp. 127-134, 2003. <https://doi.org/10.1145/781606.781627>
- [9] T. K. Dey and S. Goswami, “Provable surface reconstruction from noisy samples,” Comput. Geom., Vol.35, Nos.1-2, pp. 124-141, 2006. <https://doi.org/10.1016/j.comgeo.2005.10.006>
- [10] N. Amenta, S. Choi, and R. K. Kolluri, “The power crust,” Proc. 6th ACM Symp. Solid Model. Appl. (SMA’01), pp. 249-266, 2001. <https://doi.org/10.1145/376957.376986>
- [11] H. Edelsbrunner and E. P. Mücke, “Three-dimensional alpha shapes,” ACM Trans. Graph., Vol.13, No.1, pp. 43-72, 1994. <https://doi.org/10.1145/174462.156635>
- [12] C. L. Bajaj, F. Bernardini, and G. Xu, “Automatic reconstruction of surfaces and scalar fields from 3D scans,” Proc. 22nd Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’95), pp. 109-118, 1995. <https://doi.org/10.1145/218380.218424>
- [13] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, “Reconstruction and representation of 3D objects with radial basis functions,” Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’01), pp. 67-76, 2001. <https://doi.org/10.1145/383259.383266>
- [14] Y. Ohtake, A. Belyaev, and H.-P. Seidel, “3D scattered data interpolation and approximation with multilevel compactly supported RBFs,” Graph. Models., Vol.67, No.3, pp. 150-165, 2005. <https://doi.org/10.1016/j.gmod.2004.06.003>
- [15] C. Shen, J. F. O’Brien, and J. R. Shewchuk, “Interpolating and approximating implicit surfaces from polygon soup,” ACM SIGGRAPH 2004, pp. 896-904, 2004. <https://doi.org/10.1145/1186562.1015816>
- [16] S. Fleishman, D. Cohen-Or, and C. T. Silva, “Robust moving least-squares fitting with sharp features,” ACM Trans. Graph., Vol.24, No.3, pp. 544-552, 2005. <https://doi.org/10.1145/1073204.1073227>
- [17] J. Manson, G. Petrova, and S. Schaefer, “Streaming surface reconstruction using wavelets,” Comput. Graph. Forum, Vol.27, No.5, pp. 1411-1420, 2008. <https://doi.org/10.1111/j.1467-8659.2008.01281.x>
- [18] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, “Point set surfaces,” Proc. Conf. Vis. (VIS’01), pp. 21-29, 2001. <https://doi.org/10.1109/VISUAL.2001.964489>
- [19] G. Guennebaud and M. Gross, “Algebraic point set surfaces,” ACM Trans. Graph., Vol.26, No.3, Article No.23, 2007. <https://doi.org/10.1145/1276377.1276406>
- [20] A. Hornung and L. Kobbelt, “Robust reconstruction of watertight 3D models from non-uniformly sampled point clouds without normal information,” Proc. 4th Eurogr. Symp. Geom. Process. (SGP’06), pp. 41-50, 2006.
- [21] Y. Ohtake, A. G. Belyaev, M. Alexa, G. Turk, and H.-P. Seidel, “Multi-level partition of unity implicits,” ACM Trans. Graph., Vol.22, No.3, pp. 463-470, 2003. <https://doi.org/10.1145/882262.882293>
- [22] Y. Nagai, Y. Ohtake, and H. Suzuki, “Smoothing of partition of unity implicit surfaces for noise robust surface reconstruction,” Comput. Graph. Forum, Vol.28, No.5, pp. 1339-1348, 2009. <https://doi.org/10.1111/j.1467-8659.2009.01511.x>
- [23] M. Kazhdan, M. Bolitho, and H. Hoppe, “Poisson surface reconstruction,” Proc. 4th Eurogr. Symp. Geom. Process. (SGP’06), pp. 61-70, 2006.
- [24] M. Kazhdan and H. Hoppe, “Screened Poisson surface reconstruction,” ACM Trans. Graph., Vol.32, No.3, Article No.29, 2013. <https://doi.org/10.1145/2487228.2487237>
- [25] R. Hanocka, G. Metzer, R. Giryes, and D. Cohen-Or, “Point2Mesh: A self-prior for deformable meshes,” ACM Trans. Graph., Vol.39, No.4, Article No.126, 2020. <https://doi.org/10.1145/3386569.3392415>
- [26] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, “DeepSDF: Learning continuous signed distance functions for shape representation,” 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 165-174, 2019. <https://doi.org/10.1109/CVPR.2019.00025>
- [27] R. Chabra, J. E. Lenssen, E. Ilg, T. Schmidt, J. Straub, S. Lovegrove, and R. Newcombe, “Deep local shapes: Learning local SDF priors for detailed 3D reconstruction,” Proc. 16th Eur. Conf. Comput. Vis. (ECCV 2020), Part 29, pp. 608-625, 2020. https://doi.org/10.1007/978-3-030-58526-6_36
- [28] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, “Occupancy networks: Learning 3D reconstruction in function space,” 2019 IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), pp. 4455-4465, 2019. <https://doi.org/10.1109/CVPR.2019.00049>
- [29] M. Berger, A. Tagliasacchi, L. M. Seversky, P. Alliez, G. Guennebaud, J. A. Levine, A. Sharf, and C. T. Silva, “A survey of surface reconstruction from point clouds,” Comput. Graph. Forum, Vol.36, No.1, pp. 301-329, 2017. <https://doi.org/10.1111/cgf.12802>
- [30] P. Prusinkiewicz, L. Mündermann, R. Karwowski, and B. Lane, “The use of positional information in the modeling of plants,” Proc. 28th Annu. Conf. Comput. Graph. Interact. Tech. (SIGGRAPH’01), pp. 289-300, 2001. <https://doi.org/10.1145/383259.383291>
- [31] S. M. Hong, B. Simpson, and G. V. G. Baranoski, “Interactive venation-based leaf shape modeling,” Comput. Virtual Worlds, Vol.16, Nos.3-4, pp. 415-427, 2005. <https://doi.org/10.1002/cav.88>
- [32] W. Wen, B. Li, B.-J. Li, and X. Guo, “A leaf modeling and multi-scale remeshing method for visual computation via hierarchical parametric vein and margin representation,” Front. Plant Sci., Vol.9, Article No.783, 2018. <https://doi.org/10.3389/fpls.2018.00783>
- [33] P. Cignoni, M. Callieri, M. Corsini, M. Dellepiane, F. Ganovelli, and G. Ranzuglia, “MeshLab: An open-source mesh processing tool,” Eurogr. Ital. Chapter Conf., pp. 129-136, 2008. <http://dx.doi.org/10.2312/LocalChapterEvents/ItalChap/ItalianChapConf2008/129-136>
- [34] C. M. Bishop, “Pattern Recognition and Machine Learning,” Springer, 2006.
- [35] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, “The ball-pivoting algorithm for surface reconstruction,” IEEE Trans. Vis. Comput. Graph., Vol.5, No.4, pp. 349-359, 1999. <https://doi.org/10.1109/2945.817351>



Name:
Yukie Nagai

ORCID:
0000-0002-7362-2703

Affiliation:
Graduate School of Systems Design, Tokyo
Metropolitan University

Address:
6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

Brief Biographical History:

2010- Researcher, RIKEN
2011- Postdoctoral Researcher, Vienna University of Technology
2012- Postdoctoral Researcher, The University of Tokyo
2012- Postdoctoral Researcher, King Abdullah University of Science and
Technology
2013- Project Assistant Professor, The University of Tokyo
2014- Assistant Professor, The University of Tokyo
2018- Associate Professor, Tokyo Metropolitan University

Main Works:

- “SegMo: CT volume segmentation using a multi-level Morse complex,” Computer-Aided Design, Vol.107, pp. 23-36, 2019.
- “Tomographic surface reconstruction from point cloud,” Computers & Graphics, Vol.46, pp. 55-63, 2015.

Membership in Academic Societies:

- The Japan Society for Precision Engineering (JSPE)
 - The Japan Society of Mechanical Engineers (JSME)
-

Name:
Hikaru Tanaya

Affiliation:
Graduate School of Systems Design, Tokyo Metropolitan University

Address:

6-6 Asahigaoka, Hino, Tokyo 191-0065, Japan

Brief Biographical History:

2020 Received B.Eng. degree from Tokyo Metropolitan University
2022 Received M.Eng. degree from Tokyo Metropolitan University
2022- Nissan Motor Co., Ltd.

Main Works:

- “Investigating the relationship between geometric features of voids and textures in aerated foods,” J. of Advanced Mechanical Design, Systems, and Manufacturing, Vol.16, No.6, Article No.22-00112, 2022.

Membership in Academic Societies:

- The Japan Society for Precision Engineering (JSPE)
-