Elastic Shape Analysis of Tree-like 3D Objects using Extended SRVF Representation

Guan Wang, Hamid Laga, Anuj Srivastava *Fellow, IEEE*

Abstract—How can one analyze detailed 3D biological objects, such as neuronal and botanical trees, that exhibit complex geometrical and topological variation? In this paper, we develop a novel mathematical framework for representing, comparing, and computing geodesic deformations between the shapes of such tree-like 3D objects. A hierarchical organization of subtrees characterizes these objects – each subtree has a main branch with some side branches attached – and one needs to match these structures across objects for meaningful comparisons. We propose a novel representation that extends the Square-Root Velocity Function (SRVF), initially developed for Euclidean curves, to tree-shaped 3D objects. We then define a new metric that quantifies the bending, stretching, and branch sliding needed to deform one tree-shaped object into the other. Compared to the current metrics such as the Quotient Euclidean Distance (QED) and the Tree Edit Distance (TED), the proposed representation and metric capture the full elasticity of the branches (*i.e.,* bending and stretching) as well as the topological variations (*i.e.,* branch death/birth and sliding). It completely avoids the shrinkage that results from the edge collapse and node split operations of the QED and TED metrics. We demonstrate the utility of this framework in comparing, matching, and computing geodesics between biological objects such as neuronal and botanical trees. We also demonstrate its application to various shape analysis tasks such as **(i)** symmetry analysis and symmetrization of tree-shaped 3D objects, **(ii)** computing summary statistics (means and modes of variations) of populations of tree-shaped 3D objects, **(iii)** fitting parametric probability distributions to such populations, and **(iv)** finally synthesizing novel tree-shaped 3D objects through random sampling from estimated probability distributions.

Index Terms—Tree-shape space, elastic geodesics, elastic metrics, 3D shape variability, 3D tree synthesis, symmetry analysis, symmetrization, 3D atlas, tree classification, square-root velocity function (SRVF), correspondence, registration, topological variability.

✦

1 INTRODUCTION

TREE-LIKE 3D objects are ubiquitous, especially in delivery systems such as vascular systems, airway trees, blood vessels in the eye, plant roots and shoots, and neuronal structures in the brain. Analyzing and understanding the 3D structural variability of such tree-like biological objects are of great interest from multiple scientific perspectives. Shapes of objects both constrain and enable their functionalities in larger biological systems. Thus, a mathematical characterization of shapes can help provide insights into objects' functional roles in such biological phenomena as genesis, growth, and disease. For instance, studies of changes in the 3D structure of plant roots could improve their water and nutrient uptake efficiency. Several papers have also shown that neuronal morphology, *i.e.,* their types, geometry, and topology, and their connections are the key to discerning how neurons integrate information and subsequently explaining brain activity and its functions [1], [2], [3]. Also, statistical analysis of neuron morphology is crucial to understanding brain functionality and characterizing cognitive health. Alterations in neuron morphology

are not only due to a normal aging [4] but can also be the consequence of a pathology, *e.g.,* senile dementia [5] and Alzheimer disease [6].

Existing techniques for modeling shape variability are mostly limited to objects with fixed topology, *i.e.,* objects that only bend and stretch [7], [8], [9], [10], [11]. Tree-shaped objects, however, are more challenging because they exhibit variability in **(1)** their geometry, in terms of the shape of the individual branches, *e.g.,* axons and dendrites in neuronal structures, which can bend and stretch, and **(2)** topology, in terms of the structural relationships between those branches. One consequence of variable geometry and topology is that finding an optimal registration, *i.e.,* putting in correspondence parts (points, curves, and branches) across such objects, becomes a challenging problem. Recent works that explored this problem, *e.g.,* [12], [13], [14], [15], are limited to simple trees that are composed of a few branches due to their computational complexity and the NP-hard nature of the correspondence problem. Wang *et al.* [16], [17] extended these frameworks to handle complex botanical trees by pre-computing the correspondences. As such, correspondences and geodesics are computed separately, using different optimality criteria, which can be sub-optimal. Both works use metrics such as the Tree Edit Distance (TED) or the Quotient Euclidean Distance (QED), where topological changes are modeled by edge collapse and node split operations. Consequently, geodesics exhibit significant shrinkage, especially between trees that significantly bend (see Fig. 1-(a)) and between trees with large topological differences (see Fig. 1-(b)).

Duncan *et al.* [18] addressed this problem by represent-

• *Guan Wang is with Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China, Huzhou, 313000, China. Email: wangguan12621@gmail.com.*
• *Hamid Laga is with the School of Information Technology, the Centre of Biosecurity and One Health (Health Futures Institute), and the Centre for Healthy Ageing (Harry Butler Institute), Murdoch University. Email: H.Laga@murdoch.edu.au*
• *Anuj Srivastava is with the Department of Statistics, Florida State University. Email: anuj@stat.fsu.edu*
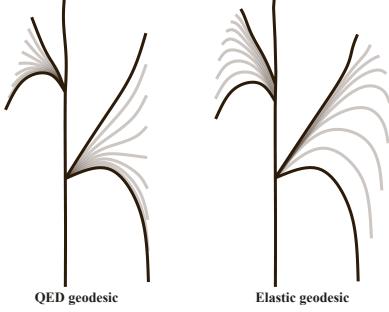
(a) Geodesic shrinkage between trees that significantly bend.

QED geodesic

Elastic geodesic
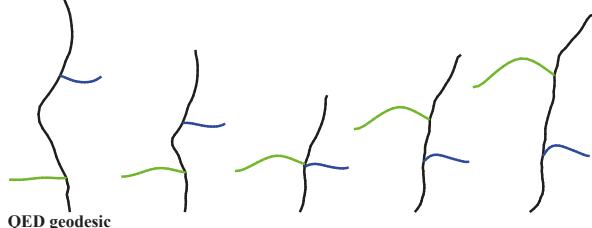
(b) Geodesic shrinkage when trees exhibit different topologies. Branch correspondences are color-coded.

Fig. 1: Examples of geodesics computed using the QED metric and with the proposed framework. Observe that when using the QED metric, **(a)** the branches shrink along the geodesic in the presence of large bending, and (b) the trees along the geodesic shrink in the presence of topological differences.

ing 2D tree-shaped objects using curves that meet at the bifurcation points. While this representation is efficient, it is limited to simple 2D neuronal trees, *i.e.,* trees composed of one main branch and some lateral branches. In this paper, we generalize the framework of Duncan *et al.* [18] to complex and arbitrary tree-shaped 3D objects. We propose a novel recursive representation where a tree-shaped object is characterized by a hierarchical organization of subtrees, *i.e.,* some side branches attached to the main branch. The task of comparing and registering two individual branches requires an optimal bending and stretching of one branch to align it onto the other. The task of comparing and registering two subtrees, each composed of one main branch and multiple side branches, is complex. It requires: **(i)** optimally sliding branches along the main branch to capture the topological differences between the two subtrees, and **(ii)** bending and stretching the branches to align them. By defining a metric that quantifies these deformations, *i.e.,* bending, stretching, and branch sliding, one can specify geodesic deformations between two subtrees as the shortest path under this metric, after factoring out shape-preserving transformations such as translation, scale, rotation, and re-parameterization. (Note that parameterization defines correspondence [8], [19].) The

geodesic length between two tree-shaped objects is then a measure of dissimilarity between the two objects. Defining this metric recursively across the entire hierarchy of the tree-shaped objects allows us to analyze complex tree-shaped 3D objects.

In this paper, we develop this mathematical framework for deriving tools that:

- Compute correspondences and geodesic deformations between tree-like 3D objects even when undergoing significant bending, stretching, and complex topological deformations.
- Compute statistical atlases, *i.e.,* means and principal modes of variation, of collections of tree-like 3D objects.
- Characterize the geometric and structural variability within a collection using probability distributions.
- Synthesize 3D tree-like structures using random sampling.

We demonstrate these tools using datasets such as 3D botanical trees and neuronal structures [20], [21], [22].

The remainder of this paper is organized as follows; After reviewing the related work in Section 2, we introduce a novel tree-shape space (Section 3) and a metric on this space (Section 4) to quantify bending, stretching, and topological changes in tree-shaped 3D objects. We then develop algorithms for computing one-to-one correspondences and geodesics, *i.e.,* optimal deformation paths with respect to the chosen metric, between tree-shaped 3D objects (Section 5). Using these building blocks, we develop computational tools for computing statistical summaries of collections of tree-shaped 3D objects (Section 6) and for synthesizing novel tree-shaped 3D models (Section 7). Finally, we present the results in Section 8 and conclude in Section 9.

2 RELATED WORK

Statistical shape analysis has been studied extensively by the computer vision, computer graphics, and statistics communities. There are two subproblems, which are essential for statistical shape analysis: **registration** and **optimal deformation**. Registration is the problem of finding a one-to-one matching of points across objects, *i.e.,* deciding which point on one object matches which point on the other. Optimal deformation, or geodesics, is the problem of finding an optimal continuous sequence of shapes starting from one shape and ending at the other. The optimality is measured with respect to a physically motivated metric.

There is a large body of literature that investigates these problems. Many papers are restricted to 3D objects that can be studied by bending and stretching. The idea is to treat shapes as points in a shape space equipped with a proper metric that measures the amount of bending and stretching that the shapes undergo. Equipping the space with a proper metric allows for comparing objects based on their shapes, computing geodesics, and performing statistical analysis, including regressions and shape synthesis. The seminal work from Kendall's school [23], [24], [25], [26] represents shapes with point sets that are already registered and focused only on deformations. Other approaches such

as medial surfaces [27], [28] and level sets [29] either pre-sume registration or solve for it using some independent pre-processing criterion such as the Minimum Description Length (MDL) [30]. Kilian *et al.* [31] represent surfaces by discrete triangulated meshes and compute geodesic defor-mation paths between them while assuming that the meshes are registered. Heeren *et al.* [32] propose a method for com-puting geodesic-based deformations of thin shell shapes, with extensions for computing summary statistics in the shell space [33], but with known registration.

Some other papers solve for registration using shape de-scriptors while ignoring deformation; see [34], [35] for a de-tailed survey on the topic. These methods sample (densely or sparsely) key points on the surface of the objects, compute shape descriptors at those locations, and then match the descriptors across the surfaces by minimizing an objective function composed of a data term and a set of regularization terms that impose constraints on the correspondence. How-ever, feature points and shape descriptors at these feature points represent only partial information about the shape of objects. Also, since the mapping from shape to descriptors is often not one-to-one and often not invertible, statistics on descriptors (*e.g.,* the mean of two shape descriptors) do not correspond to the statistics on shapes (*e.g.,* the mean of the corresponding two shapes).

Recent works, on the other hand, tried to address the registration and optimal deformation problems jointly. Cen-tral to this problem is the definition of a Riemannian met-ric (on the space of parameterized objects) preserved by the action of the relevant transformations, typically repa-rameterizations, translations, rotations, and perhaps scale. In particular, Srivastava *et al.* [19] introduced a particular elastic metric in conjunction with a representation called the Square-Root Velocity Function (SRVF). It has been used for joint registration and geodesic computation between planar shapes and between curves in $\mathbb{R}^d, d \geq 3$ [9]. It has been later extended to the analysis of parameterized surfaces [8], [10], [36], [37]. These methods, however, are limited to 3D models with fixed topology. They cannot capture and model topological variabilities such as those present in plant roots, botanical trees, and neuronal structures.

Closer to this paper are the techniques based on tree statistics. The seminal work of Billera *et al.* [38] proposed the notion of continuous tree-space and its associated tools for computing summary statistics. Some variants of this idea were developed for the statistical analysis of tree-structured data, *e.g.,* [39], [40]. However, these works only consider the topological structure of trees and ignore the geometric attributes of edges, limiting their usage. As a result, several papers have defined a more general tree-space. Examples include Feragen *et al.'s* framework [12], [13], [14], [15], which proposed a tree-shape space for com-puting statistics of airway trees, and its extension to com-plex botanical trees [16], [17]. Despite their efficiency and accuracy in certain situations, these techniques exhibit three main fundamental limitations. **First**, they use the Quotient Euclidean Distance (QED) where the geodesic between two branches is equivalent to a linear interpolation between them. As such, the QED is not suitable for capturing large elastic deformations, *i.e.,* bending and stretching, of the branches. This is illustrated in Fig. 1-(a), which shows
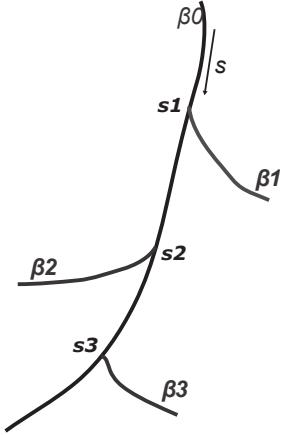

Fig. 2: The representation of a simple tree shape.

an example of a geodesic between two simple trees that are in correspondence but their branches significantly bend. Observe that under the QED metric, the branches unnatu-rally shrink along the geodesic but not with the proposed elastic geodesic. **Second**, they represent tree shapes as a *father-child branching* structure, which leads to significant shrinkage along the geodesics between trees that exhibit large topological differences; see Fig. 1-(b). In fact, the QED metric involves node split and edge collapse operations. However, edge collapse operations result in the unnatural shrinkage of the trees along the geodesic, *i.e.,* the trees become small (due to edge collapse) before growing back due to node split. This shrinkage will be significant if the source and target trees significantly differ in topology, since computing the geodesic between them would require a large number of edge collapse and node split operations; please refer also to the third row of Fig. 6. **Third**, branch-wise correspondences need to be manually specified, especially when dealing with complex tree-like structures.

In contrast to the state-of-the-art, we propose in this paper a statistical framework that is more suitable for an-alyzing complex tree-shaped 3D objects such as botanical trees, plant roots, and neuronal structures. It builds upon and extends the recent work of Duncan *et al.* [18], which showed that *main-side branching* representation is more ef-ficient for capturing topological changes. To the best of our knowledge, this is the first approach that deals with the statistical modeling of such complex tree-shaped 3D objects. In addition to classical applications such as the classification of tree-shaped 3D objects based on their shape and the computation of correspondences and geodesics, we demonstrate through experiments that this framework can produce reasonable statistical summaries. It also enables the synthesis of tree-shaped 3D objects, either randomly or in a controlled manner, and thus can be used to populate vir-tual environments and generate simulated data for training deep neural networks, *e.g.,* for the 3D reconstruction from monocular RGB images [41].

3 REPRESENTATION

The input to our framework is a collection of tree-like 3D objects such as plant roots, botanical trees, or neuronal

structures. We first skeletonize each object and convert it into a set of curves, one for each branch. The skeleton of each branch can now be seen as a curve of the form $f : [0,1] \to \mathbb{R}^3$ such that $f(s) = (x(s), y(s), z(s))$. The parameter $s \in [0,1]$ is expressed as the proportion of arc length along the skeletal curves of β. Let $\mathcal{A}_f$ be the space of all such curves. We then augment the skeletal curves with additional attributes such as the thickness at each skeletal point of the branch it represents. Each branch β can then be seen as a continuous curve in $\mathbb{R}^3 \times \mathbb{R}^+$ of the form:

$$\begin{aligned} \beta &: [0,1] \to \mathbb{R}^3 \times \mathbb{R}^+, \\ \beta(s) &\equiv (f(s), r(s)) = (x(s), y(s), z(s), r(s)). \end{aligned} \quad (1)$$

Here, $\mathbb{R}^+$ is the set of strictly positive real numbers, $f(s) = (x(s), y(s), z(s))$ is the 3D coordinates of the skeletal curve at s, and $r(s)$ is its thickness.

We organize the branches into layers, with layer zero representing the main branch. With this setup, a tree-like object can be represented recursively by writing $\boldsymbol{\beta} = (\beta^0, \{\boldsymbol{\beta}^i, s^i\}_{i=1}^n)$, where:

- $\beta = \beta^0$ is the main branch, *e.g.*, the trunk in the case of a botanical tree.
- $\boldsymbol{\beta}^i$ is a subtree attached to β^0 at the bifurcation point $\beta^0(s^i)$ with $s^i \in [0,1]$.

This is illustrated in Figure 2. If a subtree $\boldsymbol{\beta}^i$ further contains subtrees, then that can be represented recursively by repeating this idea at every level.

To make the representation translation and scale invariant, we first translate each tree so that the start point of its main branch is located at the origin, and then scale the entire tree so that the length of its main branch is one. Note that invariance to scale is optional as it may not be required for some applications such as growth analysis. In what follows, we assume that all the objects have been normalized for translation and scale, and thus they are elements of a pre-shape space[1] of trees, hereinafter referred to as pre-tree shape space and denoted by $\mathcal{C}_{\boldsymbol{\beta}}$. It can be recursively defined as follows.

- Let $\mathcal{C}_{\boldsymbol{\beta}} = \mathcal{C}_{\boldsymbol{\beta}}^0$ be the pre-tree shape space of all trees that only have the main branch, a single curve. We will call such structures a level-0 tree. It can be shown that

$$\mathcal{C}_{\boldsymbol{\beta}}^0 = \mathcal{A}_f \times \mathcal{T}, \quad (2)$$

  where $\mathcal{T}$ is the space of functions of the form $[0,1] \to \mathbb{R}^+$, and represents the thickness of the branch at each point along its skeleton. Note that $\mathcal{C}_{\boldsymbol{\beta}}^0$ is in fact the space of all curves defined by Eqn. (1).
- Let now $\mathcal{C}_{\boldsymbol{\beta}}^k, k \geq 1$ be the pre-tree shape space of all trees that can have up to k levels. This space can be recursively defined as follows;

$$\mathcal{C}_{\boldsymbol{\beta}}^k = \mathcal{C}_{\boldsymbol{\beta}}^0 \times \bigsqcup_{n=0}^{\infty} \left(\mathcal{C}_{\boldsymbol{\beta}}^{k-1} \times [0,1] \right)^n. \quad (3)$$

  This means that a level-k tree can have n structures attached to the main branch, each structure being a level-$(k-1)$ tree, at specified attachment points.

1. A pre-shape space of curves, surfaces, or trees is the space of all such curves, surfaces, or trees before removing or factoring out rotations and reparameterizations, which are shape-preserving transformations.

4 METRIC FOR TREE-SHAPES

In order to perform statistical shape analysis and modeling of the geometry and topology of a collection of tree shapes, we need to define a distance metric in the shape space of trees and a mechanism for computing correspondences and geodesics between elements in that space. The metric needs to quantify shape-changing deformations, *i.e.*, bending, stretching, and topological changes, while being invariant to shape-preserving transformations, *i.e.*, translations, global scaling, and rotations.

4.1 The elastic metric for the shape of branches

We propose to use an elastic metric that captures the bending and stretching of the branches. Consider a path $\alpha_{\beta_1 \to \beta_2} : [0,1] \to \mathcal{C}_{\boldsymbol{\beta}}$, which deforms β_1 onto β_2. In other words, $\alpha(0) = \beta_1, \alpha(1) = \beta_2$, and $\forall t \in [0,1], \alpha(t) \in \mathcal{C}_{\boldsymbol{\beta}}$. Its length $L[\alpha]$ is given by:

$$L[\alpha] = \int_0^1 \langle\langle \dot{\alpha}(t), \dot{\alpha}(t) \rangle\rangle^{\frac{1}{2}} dt, \quad (4)$$

where $\dot{\alpha} = \frac{\partial \alpha(t)}{\partial t}$ is the velocity vector, and $\langle\langle \cdot, \cdot \rangle\rangle$ is a certain metric. Instead of using the $\mathbb{L}^2$ metric as in [14], [16], we use a metric that quantifies the bending and stretching of the branches.

Physically, bending can be quantified by measuring changes in the orientation of the tangent vectors to the skeletal curve f of β along the deformation path. Stretching can be decomposed into two components; the first one is related to the elongation of the skeletal curve f, which can be quantified by looking at how the magnitude of the tangent vector to f at every point s changes along the deformation path. The second component is related to changes in the thickness of the branch and can be measured by looking at how the thickness r at each point s varies along the deformation path. Let us write:

$$f'(s) = \frac{\partial f(s)}{\partial s} = \theta(s)e^{\phi(s)}, \quad (5)$$

where $\theta(s)$ is the unit tangent vector to f at s, and $e^{\phi(s)}$ is the magnitude of the tangent vector to f at s. (We also refer to $e^{\phi(s)}$ as the speed.) In other words:

$$e^{\phi(s)} = \|f'(s)\| = \left\| \frac{\partial f(s)}{\partial s} \right\|, \text{ and } \theta(s) = \frac{f'(s)}{\|f'(s)\|}. \quad (6)$$

One can then define an elastic metric as the weighted sum of the changes of ϕ, θ, and r along the path α. In other words:

$$\begin{aligned} \langle\langle \dot{\beta}, \dot{\beta} \rangle\rangle &\equiv a \int \langle \dot{\theta}(s), \dot{\theta}(s) \rangle e^{\phi(s)} ds + \\ & b \int \dot{\phi}(s)^2 e^{\phi(s)} ds + c \int \dot{r}(s)^2 e^{\phi(s)} ds. \end{aligned} \quad (7)$$

Here, $\dot{g}$ denotes the derivative of the function g with respect to t. The first term of Eqn. (7) quantifies bending by measuring changes in the orientation of the tangent vector to the skeletal curve. The second and third terms quantify stretching. The first two terms are equivalent to the elastic metric between curves in $\mathbb{R}^3$, which has been introduced by Srivastava *et al.* [19]. While the full elastic metric of Eqn. (7) is the most appropriate for analyzing the shape of branches, it is computationally very expensive. To reduce

the computation time, Srivastava *et al.* [19] introduced the Square Root Velocity Function (SRVF) of a curve f as a mapping $\mathcal{Q}$ of the form:

$$\mathcal{Q}(f)(s) = \begin{cases} \frac{f'(s)}{\sqrt{\|f'(s)\|}} & \text{if } f'(s) \text{ exists and } \|f'(s)\| \neq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (8)$$

More importantly, Srivastava *et al.* [19] showed that the $\mathbb{L}^2$ metric in the space of SRVFs is equivalent to the elastic metric of Eqn. (7) when setting $a = 1, b = \frac{1}{4}$, and $c = 0$. In our case, we define the Extended SRVF (ESRVF) of a branch $\beta = (f, r)$ as the pair $q = \text{SRVF}([f, cr])$, with c is the weight of the third term of Eqn. (7). We can easily show that the $\mathbb{L}^2$ metric in the space of ESRVFs, hereinafter denoted by Ω, is equivalent to the elastic metric of Eqn. (7) when setting $a = c = 1, b = \frac{1}{4}$. This is very important since, instead of working with the complex elastic metric, we can represent the shape of each branch using its ESRVF and use the associated $\mathbb{L}^2$ metric to capture the bending and stretching of branches.

The ESRVF representation inherits all the properties of the SRVF. It is invariant to translations since it is defined in terms of derivatives. Also, the action of the reparameterization group on the ESRVF is by isometry. In other words, $\forall \gamma \in \Gamma, \|q_1 - q_2\| = \|q_1 \circ \gamma - q_2 \circ \gamma\|$. Here Γ is the space of all diffeomorphisms of the domain $[0, 1]$ to itself. With this, aligning two branches β_1 and β_2 becomes the problem of finding the optimal rotation $O^* \in SO(3)$ and reparametrization $\gamma^* \in \Gamma$ such that:

$$(O^*, \gamma^*) = \underset{O \in SO(3), \gamma \in \Gamma}{\text{argmin}} \|q_1 - O(q_2, \gamma)\|^2. \quad (9)$$

Here, $(q_2, \gamma) = \text{ESRVF}(\beta_2 \circ \gamma)$. Thus, $O(q_2, \gamma)$ is the process of applying the rotation O to $(q_2, \gamma) = \text{ESRVF}(\beta_2 \circ \gamma)$. The distance between the two branches is then defined as $\|q_1 - O^*(q_2, \gamma^*)\|$. Thus, the invariance to rotations is achieved through the optimization over $SO(3)$ in Eqn. (9).

With this setup, each tree shape $\boldsymbol{\beta} = (\beta^0, \{\boldsymbol{\beta}^i, s^i\}_{i=1}^n)$ will be represented with the collection of the ESRVFs of its branches. We refer to this recursive representation of the form $\boldsymbol{q} = (q^0, \{\boldsymbol{q}^i, s^i\}_{i=1}^n)$ as the Square Root Velocity Function Tree (SRVFT). The pre-tree shape space in which $\boldsymbol{q}$ resides can now be recursively defined as follows;

- Let $\mathcal{C}_{\boldsymbol{q}} = \mathcal{C}_{\boldsymbol{q}}^0$ be the pre-tree shape space of all SRVFTs that have one single branch. One can show that

$$\mathcal{C}_{\boldsymbol{q}}^0 = \mathcal{A}_q \times \mathcal{T}, \quad (10)$$

where $\mathcal{A}_q = \text{SRVF}(\mathcal{A}_f)$, *i.e.*, the image of $\mathcal{A}_f$ using the SRVF map, and $\mathcal{T}$ is the space of functions of the form $[0, 1] \to \mathbb{R}^+$ and which represent the thickness of the branch at each point along its skeleton.
- Let now $\mathcal{C}_{\boldsymbol{q}}^k, k \geq 1$ be the pre-tree shape space of all SRVFTs that can have up to k levels. This space can be recursively defined as follows;

$$\mathcal{C}_{\boldsymbol{q}}^k = \mathcal{C}_{\boldsymbol{q}}^0 \times \bigsqcup_{n=0}^{\infty} \left(\mathcal{C}_{\boldsymbol{q}}^{k-1} \times [0, 1] \right)^n. \quad (11)$$

This representation has many nice properties. For instance,

- The mapping from $\mathcal{C}_{\boldsymbol{\beta}}$ to $\mathcal{C}_{\boldsymbol{q}}$ is surjective.

- The mapping, however, is not injective, but two curves are mapped to the same SRVF space if and only if they are translates of each other.
- The inverse mapping has a closed analytical form. In other words, for a given pair (q, r), one can find, analytically and up to a translation, a branch β whose SRVF is (q, r).

As a consequence of the last property, and since the locations of the subtrees is encoded in the SRVFT representation, the inverse mapping, *i.e.*, the mapping from $\mathcal{C}_{\boldsymbol{q}}$ to $\mathcal{C}_{\boldsymbol{\beta}}$, can be found in an analytical form, up to a global translation.

4.2 The elastic metric for comparing tree-shapes

The next step is to define a metric in the space of SRVFTs and a mechanism for computing correspondences and geodesics between points in this space. We do this by generalizing the metric proposed by Duncan *et al.* [18] for the analysis of simple trees, composed of the main branch and a set of side branches, to complex 3D trees of arbitrary levels of hierarchy. Let $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ be two tree shapes represented with their SRVFTs $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in $\mathcal{C}_{\boldsymbol{q}}$. Let us first assume that the two trees have the same number of branches and are in branch-wise correspondence. We define the distance between $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ recursively as follows;

$$d_{\mathcal{C}_{\boldsymbol{q}}}^2(\boldsymbol{q}_1, \boldsymbol{q}_2) = \lambda_m \| q_1^0 - q_2^0 \|^2 + \lambda_s \sum_{i=1}^n d_{\mathcal{C}_{\boldsymbol{q}}}^2(\boldsymbol{q}_1^i, \boldsymbol{q}_2^i) + \\ \lambda_p \sum_{i=1}^n (s_1^i - s_2^i)^2. \quad (12)$$

The first term of Eqn. (12) measures the amount of bending and stretching needed to align one main branch to another. The second term, which is computed recursively using Eqn. (12), is the dissimilarity between two subtrees $\boldsymbol{q}_1^i$ and $\boldsymbol{q}_2^i$ attached, respectively, to the bifurcation points s_1^i and s_2^i. Note that when $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ are null trees, then we set $d_{\mathcal{C}}(\boldsymbol{q}_1, \boldsymbol{q}_2) = 0$. The third term is the distance between the locations of the corresponding main branches of the subtrees $\boldsymbol{q}_1^i$ and $\boldsymbol{q}_2^i$. The parameters $\boldsymbol{\lambda} = (\lambda_m, \lambda_s, \lambda_p)$ control the relative cost of deforming the main branch, deforming the subtrees connected to the main branch, and moving the positions of the subtrees connected to the main branch. Section 8.2 shows the effects of these parameters with practical examples.

4.3 Invariant metric

A proper metric for comparing the shape of tree-like objects should be invariant to shape-preserving transformations. The translation and scale have already been factored out in a pre-processing step. Now, we consider rotations, reparameterization of the branches, and permutations of the orders of the lateral subtrees attached to a branch. Let:

- $O \in SO(3)$ be a global rotation applied to the entire tree,
- $\gamma = (\gamma^0, \{\gamma^i\}_{i=1}^n)$ be the reparameterization of the main branch and its subtrees. Specifically, $\gamma^0 \in \Gamma$ is a diffeomorphism that applies to the main branch of $\boldsymbol{q}$ and $\boldsymbol{\gamma}^i$ the reparameterization, defined recursively, of the $i-$th subtree.

- $\boldsymbol{\sigma} = (\sigma^0, \{\boldsymbol{\sigma}^i\}_{i=1}^{n_k})$ such that $\sigma^0 \in \boldsymbol{\sigma}$ is the permutation of the orders of the lateral subtrees on their corresponding main branch q_k^0 (here, $\boldsymbol{\sigma}$ is the space of such permutations), and $\boldsymbol{\sigma}^i$ defines recursively these permutations for the $i-$th subtree.

Let $(\boldsymbol{q}, O, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ denote the result after applying these three transformations to $\boldsymbol{q}$. Since these transformations are shape-preserving, $\boldsymbol{q}$ and $(\boldsymbol{q}, O, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ have the same shape. They thus are equivalent under the action of global rotations, branch reparameterizations, and permutations of the indices of the lateral subtrees of the different branches. We treat $(O, \boldsymbol{\gamma}, \boldsymbol{\sigma})$ as a shape-preserving nuisance action and define a quotient space on the closures of orbits under this action. We then define the rotation, reparameterization, and index permutation-invariant distance between two tree shapes $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ as the distance between their orbits, which we denote by $[\boldsymbol{q}_1]$ and $[\boldsymbol{q}_2]$. It is defined as

$$d^2\left([\boldsymbol{q}_1], [\boldsymbol{q}_2]\right) = \inf_{\tilde{\boldsymbol{q}}_1 \in [\boldsymbol{q}_1], \tilde{\boldsymbol{q}}_2 \in [\boldsymbol{q}_2]} \{d_{\mathcal{C}_q}^2(\tilde{\boldsymbol{q}}_1, \tilde{\boldsymbol{q}}_2)\}. \quad (13)$$

In other words, to compute the geodesic distance between two equivalence classes, or two orbits in the quotient space, $[\boldsymbol{q}_1]$ and $[\boldsymbol{q}_2]$, one can select two representative instances $\boldsymbol{q}_1 \in [\boldsymbol{q}_1]$ and $\boldsymbol{q}_2 \in [\boldsymbol{q}_2]$, and then optimize over all possible rotations, branch reparameterizations, and branch order permutations of the two instances. From the computational point of view, we fix $\boldsymbol{q}_1$ and optimize over all possible rotations, branch reparameterizations, and branch order permutations of $\boldsymbol{q}_2$. In other words,

$$d^2\left(\boldsymbol{q}_1, \boldsymbol{q}_2\right) = \inf_{\substack{O \in SO(3) \\ \boldsymbol{\gamma} \in \boldsymbol{\Gamma} \\ \boldsymbol{\sigma} \in \mathbf{S}}} d_{\mathcal{C}_q}^2\left(\boldsymbol{q}_1, (\boldsymbol{q}_2, O, \boldsymbol{\gamma}, \boldsymbol{\sigma})\right), \quad (14)$$

where

$$\begin{aligned} d_{\mathcal{C}_q}^2\left(\boldsymbol{q}_1, (\boldsymbol{q}_2, O, \boldsymbol{\gamma}, \boldsymbol{\sigma})\right) = &\lambda_m \parallel q_1^0 - O(q_2^0, \gamma_0) \parallel^2 + \\ &\lambda_s \sum_{i=1}^n d^2\left(\boldsymbol{q}_1^i, \boldsymbol{q}_2^{\boldsymbol{\sigma}(i)}\right) + \\ &\lambda_p \sum_{i=1}^n \left(s_1^i - s_2^{\boldsymbol{\sigma}(i)}\right)^2. \end{aligned} \quad (15)$$

The optimal registration of $\boldsymbol{q}_2$ onto $\boldsymbol{q}_1$ can then be found by solving the following optimization problem:

$$(\tilde{O}, \tilde{\boldsymbol{\gamma}}, \tilde{\boldsymbol{\sigma}}) = \operatorname*{argmin}_{O, \boldsymbol{\gamma}, \boldsymbol{\sigma}} d_{\mathcal{C}_q}^2\left(\boldsymbol{q}_1, \boldsymbol{q}_2, O, \boldsymbol{\gamma}, \boldsymbol{\sigma}\right). \quad (16)$$

The challenge now is how to optimize simultaneously over all these transformation spaces. This will be discussed in Section 5.

4.4 Trees with different numbers of side branches

The formulation presented in Sections 4.2 and 4.3 assumes that the trees have the same number of branches and the same topology. Thus, a one-to-one branch-wise correspondence exists. However, this is not the case in practice. Two tree models rarely have the same number of bifurcation points and branches. As a result, these trees lie on disjoint subspaces, making it difficult to compare them and compute geodesics. One way to overcome this issue is by adding null branches, *i.e.,* branches of length zero.

First, we define the order of a tree $\boldsymbol{\beta}$, and subsequently of $\boldsymbol{q}$, by taking the maximum of the number of bifurcation points on its branches. Let $n_i, i = 1, 2$ be the order of the trees $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, and let $n = \max(n_1, n_2)$. Then, we add to each branch l of the $i-$th tree, $n - n_i^l$ null branches. (Here, n_i^l is the order of branch l.) By doing so, all k-level trees become elements of the same pre-tree shape space. This facilitates their comparison, and subsequently putting them in correspondence, and computing geodesic paths. The location of each additional null branch, *i.e.,* the value of its parameter s, is initialized to the s value of its initial corresponding branch. These correspondences will then get updated during the optimization over the branch-wise correspondences.

5 CORRESPONDENCES AND GEODESICS

With this formulation, the metric in the pre-tree shape space $\mathcal{C}_q$ is a weighted sum of $\mathbb{L}^2$ distances. This has significant practical benefits. In particular, instead of working in the original space of 3D objects, which is nonlinear and equipped with a complex metric, one can map the input tree-like objects into the SRVFT space, use the simple metric of Eqn. (14) to compute correspondences, geodesics, and statistics using standard tools from vector calculus, and finally map the results back to the original space of trees for visualization. Since the SRVFT mapping is one-to-one and onto (up to global translation), the inverse mapping exists, is unique, and more importantly, has a closed analytical form. This invertibility property is very important in practice.

Given two tree-shaped objects $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ represented with their SRVFTs $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$, we obtain one-to-one correspondences by solving the optimization problem of Eqn. (16). This is done by alternating the optimisation over $SO(3)$, Γ, and $\mathbf{S}$, *i.e.,*

- Optimize over $O \in SO(3)$, assuming fixed parameterizations and permutations, which is straightforward to achieve using Procrustes analysis [23]. Then,
- Optimize over $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$ and $\boldsymbol{\sigma} \in \mathbf{S}$ while assuming a fixed rotation.

In this section, we discuss the process of optimizing over $\boldsymbol{\gamma} \in \boldsymbol{\Gamma}$ and $\boldsymbol{\sigma} \in \mathbf{S}$. We first consider the simple case of trees with two levels of hierarchy, *i.e.,* trees that only have one main branch and multiple side, or lateral, branches (Section 5.1). We then show how this method can be generalized to complex tree-shaped 3D objects of arbitrary branching structures (Section 5.2). Algorithms 1 and 2 summarize the overall procedures.

5.1 Correspondence between simple tree shapes

Assume the simple case where the two trees are composed of only one main branch and some lateral branches. In this case $\boldsymbol{\gamma} = (\gamma^0, \{\gamma^i\}_{i=1}^n)$ and $\boldsymbol{\sigma} = (\sigma_0)$. Since the first term of Eqn. (15), which measures the distance between the two main branches, is independent of the other two terms, we first find, using the approach of Srivastava *et al.* [19], the optimal re-parameterization γ^0 that elastically aligns the main branch of $\boldsymbol{q}_2$ onto the main branch of $\boldsymbol{q}_1$. Next, we formulate the joint problem of **(1)** finding which side branch of $\boldsymbol{q}_2$ is matched to which branch of $\boldsymbol{q}_1$ (*i.e.,* using

---

Algorithm 1 Alignment procedure.

Input:

- $n\_levels = 2, 3$ or 4.
- q^1, q^2, the SRVFT representations of two tree-shaped objects to be aligned.
- $\lambda = (\lambda_m, \lambda_s, \lambda_p)$, the distance weight parameters.
- $n_{\max}$, the number of iterations.

Output:

- $O, \gamma,$ and σ.
- E_{final}, the distance between q^1 and q^2.

1: **procedure** ALIGNTREES(q^1, q^2, n_levels)
2:    $(\gamma, \sigma) =$ ReparamPermute($O_{\text{id}}, q^1, q^2, \lambda, n\_$levels);
3:    **for** $n_{\max}$ iterations **do**
4:        $O =$ Procurstes($q^1, (q^2, \gamma, \sigma), \lambda$).
5:        $(\gamma, \sigma) =$ ReparamPermute($O, q^1, q^2, \lambda, n\_$levels);
6:    **end for**
7:    $E_{\text{final}} = d_{\mathcal{C}_q}(q^1, O(q^2, \gamma, \sigma))$
8:    Return $(O, \gamma, \sigma, E_{\text{final}})$.
9: **end procedure**

---

the third term of Eqn. (15)) and **(2)** elastically registering the corresponding branches (*i.e.,* using the second term of Eqn. (15)), as a linear assignment problem. That is, we build a pairwise distance matrix E of size $n_1 \times n_2$ where

$$E_{ij} = \lambda_s \inf_{O, \gamma} \|q_1^i - O(q_2^j, \gamma)\|^2 + \lambda_p \left(s_1^i - s_2^j \right)^2. \quad (17)$$

We then apply the Kuhn-Munkres algorithm, also known as the Hungarian Algorithm [42], to compute the optimal matching σ_0. Next, for each matching branches q_1^i and $q_2^{\sigma_0(i)}$, we compute the reparameterization γ_i as

$$(O_i, \gamma_i) = \underset{O, \gamma}{\operatorname{argmin}} \|q_1^i - O(q_2^{\sigma_0(i)}, \gamma)\|^2 \quad (18)$$

by using the approach of Srivastava *et al.* [19]. Note that, in contrast to Duncan *et al.* [18], which builds an $(n_1 + n_2) \times (n_1 + n_2)$ cost matrix, we solve the linear assignment problem by building a $\max(n_1, n_2) \times \max(n_1, n_2)$ cost matrix. In this way, we reduce the worst-case time complexity from $O((n_1 + n_2)^3)$ to $O(\max(n_1, n_2)^3)$ and reduce the computation time by a factor of 8. This is important especially when processing recursively tree shapes with many layers.

Note that, when solving the optimization of Eqn. (16), the relative cost of deforming the main branch, deforming the subtrees connected to the main branch, and moving the positions of the subtrees connected to the main branch is controlled by the parameters $\lambda = (\lambda_m, \lambda_s, \lambda_p)$. The parameter λ_m quantifies bending and stretching of the main branch. λ_s allows the recursion over the subtrees and quantifies the similarity between subtrees. λ_p quantifies branch sliding. In summary,

- High values of λ_p and low values of λ_m will penalize branch sliding. Thus, the algorithm will try to match side branches that are close to each other. It will prioritize correspondences that will result in bending and stretching over those that will cause changes in topology.

- Low values of λ_p and high values of λ_m will have the opposite effect.
- The parameter λ_s controls the contribution, or importance, of the subtrees to the overall distance measure.

Section 8.2 shows, with examples, the effect of different values of these parameters.

5.2 Extension to complex tree shapes

To handle complex trees, one can recursively run the procedure described in Section 5.1 over the entire hierarchy of the trees. By doing so, however, the computation time would increase exponentially. We approximate this procedure by considering three levels at a time. That is, we first consider the main branch and the subtrees attached to it. However, for each subtree, we only consider its main branch and the side branches attached to it. To align two trees, we first align their main branches, compute the assignment matrix where each entry E_{ij} is the optimal distance between the subtree q_1^i and q_2^j. Since these two subtrees have only two levels, E_{ij} can be computed using the procedure described in Section 5.1. Similar to Section 5.1, we apply the Hungarian algorithm to find the assignment σ_0 which matches subtrees on q_2 onto subtrees on q_1. By repeating this procedure recursively on q_1^i and $q_2^{\sigma_0(i)}$, we obtain a full matching between the complex trees. Algorithms 1 and 2 summarize the overall procedure.

5.3 Computing geodesics

Let $\tilde{q}_2 = (q_2, \tilde{O}, \tilde{\gamma}, \tilde{\sigma})$. Since the metric is a weighted Euclidean norm of $\mathbb{L}^2$ distances, the geodesic $\tilde{\alpha}$ between q_1 and q_2 is a straight line that connects q_1 to $\tilde{q}_2$, *i.e.,* :

$$\tilde{\alpha}(t) = (1 - t)q_1 + t\tilde{q}_2, t \in [0, 1]. \quad (19)$$

Finally, for visualization, we map $\alpha(t), t \in [0, 1]$, back to the nonlinear space of tree shapes $\mathcal{C}_\beta$ using the inverse SRVF mapping, which has a closed analytical form; see [19].

6 STATISTICS ON THE TREE-SHAPE SPACE

The ability to compute correspondences and geodesics between 3D tree-shaped objects enables a wide range of shape analysis tasks. In this section, we show how these fundamental tools can be used to compute the means and modes of variability of a collection of tree-shaped 3D objects $\{\beta_i, i = 1, \ldots, m\}$, *e.g.,* botanical trees, plant roots, or neuronal structures. Let $\{q_i, i = 1, 2, \ldots, m\}$ be their corresponding SRVFT representations. To compute the mean tree μ, we first compute the mean SRVFT representation μ_q and then map it back to the space of tree shapes.

Mathematically, the mean μ_q can be regarded as the point in $\mathcal{C}_q$ that is as close as possible to all the tree shapes in $\{q_i, i = 1, \ldots, m\}$. The closeness is defined with respect to the metric of Eqn. (15). In other words,

$$\mu_q = \underset{\substack{O_i \in SO(3) \\ \gamma_i \in \Gamma, \sigma_i \in \mathbf{s} \\ i = 1 \ldots m}}{\operatorname{argmin}} \sum_{i=1}^{m} d_{\mathcal{C}_q}^2(q, O(q_i, \gamma_i, \sigma_i)). \quad (20)$$

Solving Eqn. (20) involves finding μ_q, also known as the Karcher mean, while simultaneously registering every q_i

Algorithm 2 Reparameterization and permutation.

- $n\_levels = 2, 3$ or 4.
- $\boldsymbol{q}^1, \boldsymbol{q}^2$, the SRVFT representations of two tree-shaped objects to be aligned.
- $\boldsymbol{\lambda} = (\lambda_m, \lambda_s, \lambda_p)$, the distance weight parameters.
- n_1, n_2, the number of side branches in $\boldsymbol{q}^1$, respectively, $\boldsymbol{q}^2$.
- **DynamicProgQ** denotes the dynamic programming algorithm, as in Srivastava et al. [19], to compute the optimal reparameterization between two SRVF curves.
- **LAPJV** denotes to solve the **L**inear **S**um **A**ssignment **P**roblem using the algorithm of **J**onker and **V**olgenant [43].

Output:

- $\boldsymbol{\gamma}$ and $\boldsymbol{\sigma}$.

1: **procedure** REPARAMPERMUTE($O, \boldsymbol{q}^1, \boldsymbol{q}^2, \boldsymbol{\lambda}, n\_levels$)
2:     $N = \max(n_1, n_2)$.
3:     **for** $i = 1 : N, j = 1 : N$ **do**
4:         **if** $n\_levels = 3$ or 4 **then**
5:             $(O, \boldsymbol{\gamma}_{ij}, \boldsymbol{\sigma}_{ij}, E_{ij}) = $ AlignTrees($\boldsymbol{q}_i^1, \boldsymbol{q}_j^2$,
                                            $n\_levels - 1$).
6:         **else**
7:             $\gamma_{ij} = $ DynamicProgQ(q_i^1, Oq_j^2)
8:             $E_{ij} = \lambda_m \|q_0^1 - O(q_0^2, \gamma_{ij})\|^2 + \lambda_p (s_i^1 - s_j^2)^2$.
9:         **end if**
10:     **end for**
11:     $\boldsymbol{\sigma}_0 = $ LAPJV(E).
12:     $\gamma_0 = $ DynamicProgQ (q_0^1, q_0^2).
13:     **for** $i = 1 : N, j = 1 : N$ **do**
14:         **if** $n\_levels = 3$ or 4 **then**
15:             $(\boldsymbol{\gamma}_i, \boldsymbol{\sigma}_i) = $ AlignTrees($\boldsymbol{q}_i^1, \boldsymbol{q}_{\boldsymbol{\sigma}_0(i)}^2, n\_levels$).
16:             $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} \cup \{\boldsymbol{\gamma}_i\}$.
17:             $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} \cup \{\boldsymbol{\sigma}_i\}$.
18:         **else**
19:             $\boldsymbol{\gamma} \leftarrow \boldsymbol{\gamma} \cup \gamma_{i\boldsymbol{\sigma}_0(i)}$.
20:             $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} \cup \{\boldsymbol{\sigma}_0\}$.
21:         **end if**
22:     **end for**
23:     Return $\boldsymbol{\gamma}, \boldsymbol{\sigma}$.
24: **end procedure**

on $\boldsymbol{\mu}_q$. This can be efficiently done via a gradient descent approach. That is:

1) Set $\boldsymbol{\mu}_q = \boldsymbol{q}_1$.
2) For $i = 1 : m$

    - Optimally register $\boldsymbol{q}_i$ onto $\boldsymbol{\mu}_q$ using Eqn. (16).
    - Let $(\tilde{O}_i, \tilde{\boldsymbol{\gamma}}_i, \tilde{\boldsymbol{\sigma}}_i)$ be the solution to Eqn. (16).

3) Set $\boldsymbol{\mu}_q = \frac{1}{m} \sum_{i=1}^m \tilde{O}_i(\boldsymbol{q}_i, \tilde{\boldsymbol{\gamma}}_i, \tilde{\boldsymbol{\sigma}}_i)$.
4) Repeat steps 2 and 3 until convergence.
5) Return $\boldsymbol{\mu}_q$ and $(\tilde{O}_i, \tilde{\boldsymbol{\gamma}}_i, \tilde{\boldsymbol{\sigma}}_i)_{i=1}^m$.

Finally, the mean tree $\boldsymbol{\mu}$ is obtained by mapping $\boldsymbol{\mu}_q$ back to the space $\mathcal{C}_\beta$. This mean shape characterizes the primary morphological properties of shapes in the collection $\{\boldsymbol{\beta}_i, i = 1, \ldots, m\}$. In what follows, let $\boldsymbol{q}_i = \tilde{O}_i(\boldsymbol{q}_i, \tilde{\boldsymbol{\gamma}}_i, \tilde{\boldsymbol{\sigma}}_i)$ be the

SRVFT representation of $\boldsymbol{\beta}_i$ but optimally registered onto the mean $\boldsymbol{\mu}_q$.

Let $\boldsymbol{v}_i = \boldsymbol{q}_i - \boldsymbol{\mu}$. The covariance matrix is then defined as $K = \dfrac{1}{m-1} \sum_{i=1}^m \boldsymbol{v}_i \boldsymbol{v}_i^t$. Its leading eigenvectors Λ_i define the principal directions of variations. The eigenvalue λ_i that corresponds to the i−th mode of variation defines the variance along the i−th direction.

With this setup, each SRVFT sample $\boldsymbol{q}$ can be modeled as a linear combination of the k leading eigenvectors:

$$\boldsymbol{q} = \boldsymbol{\mu}_q + \sum_{i=1}^k a_i \sqrt{\lambda_i} \Lambda_i, a_i \in \mathbb{R}, \quad (21)$$

Note that with this representation, we are fitting a Gaussian distribution to the input data $\{\boldsymbol{q}_i, i = 1, \ldots, m\}$. In general, however, we can fit any arbitrary distribution from the parametric or non-parametric families. Note , however, that a Gaussian distribution in the SRVFT space does not necessarily correspond to a Gaussian in the original space of trees (which is nonlinear). In fact, the mapping of distributions across the two spaces remains to be studied and this will be explored in future work.

Note also that the SRVFT space is flat only in the case when the shapes are not normalized for scale. However, the shape (quotient) space is not flat and the minimizer of Eqn. (20) may not be unique but a set of distinct shapes. The algorithm above tries to find a minimizer, which we then use to define covariance and PCA. Since the algorithm for computing the mean is a gradient-based search, it may not find the global minimizer. This is a limitation of the current approach and some recent papers have discussed the usage of stochastic searchers for reaching the global minimizer, but at the expense of a significantly higher computation time. As the objective function is bounded by zero from below, and the iterations are monotonic, the algorithm is guaranteed to converge, although it may converge to a local minimum.

7 GENERATIVE MODEL OF 3D TREE-SHAPES

We characterize the shape variability of an input population of tree shapes by fitting to the shape population a multivariate Gaussian of mean $\boldsymbol{\mu}_q$ and a diagonal covariance matrix $\mathcal{K}$ whose diagonal elements are the eigenvalues λ_i. A random 3D tree-shape can then be generated by randomly sampling from this multivariate Gaussian. That is, we first randomly sample k real numbers $a_1, \ldots, a_k \sim \mathcal{N}(0, 1)$. The SRVFT $\boldsymbol{q}$ of a random tree-shape $\boldsymbol{\beta}$ is then given by Eqn. (21), and the random tree-shape $\boldsymbol{\beta}$ is obtained by mapping $\boldsymbol{q}$ back to the space of 3D tree-shapes. In this paper, we only consider the k-leading eigenvalues such that $k \leq m$ and $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i} > 0.99$. To ensure that the synthesized tree-shapes are plausible, one can restrict a_i to be within a certain range. For example, by setting $a_i \in [-1, 1]$, the generated random trees are restricted to be within one standard deviation around the mean tree.

8 RESULTS AND DISCUSSION

In this section, we demonstrate the performance of the proposed framework in finding correspondences and computing geodesics between pairs of tree-like 3D objects (Section 8.1), the analysis of the symmetry and symmetrization

(a) Geodesic length $d_s = 29.3$. Computation time: 60.1s for computing the correspondences and 0.4s for the geodesics.

(b) Geodesic length $d_s = 31.7$. Computation time: 173.8s for computing the correspondences and 0.47s for the geodesics.

(c) Geodesic length $d_s = 121.8$. Computation time: 563.5s for computing the correspondences and 1.1s for the geodesics.
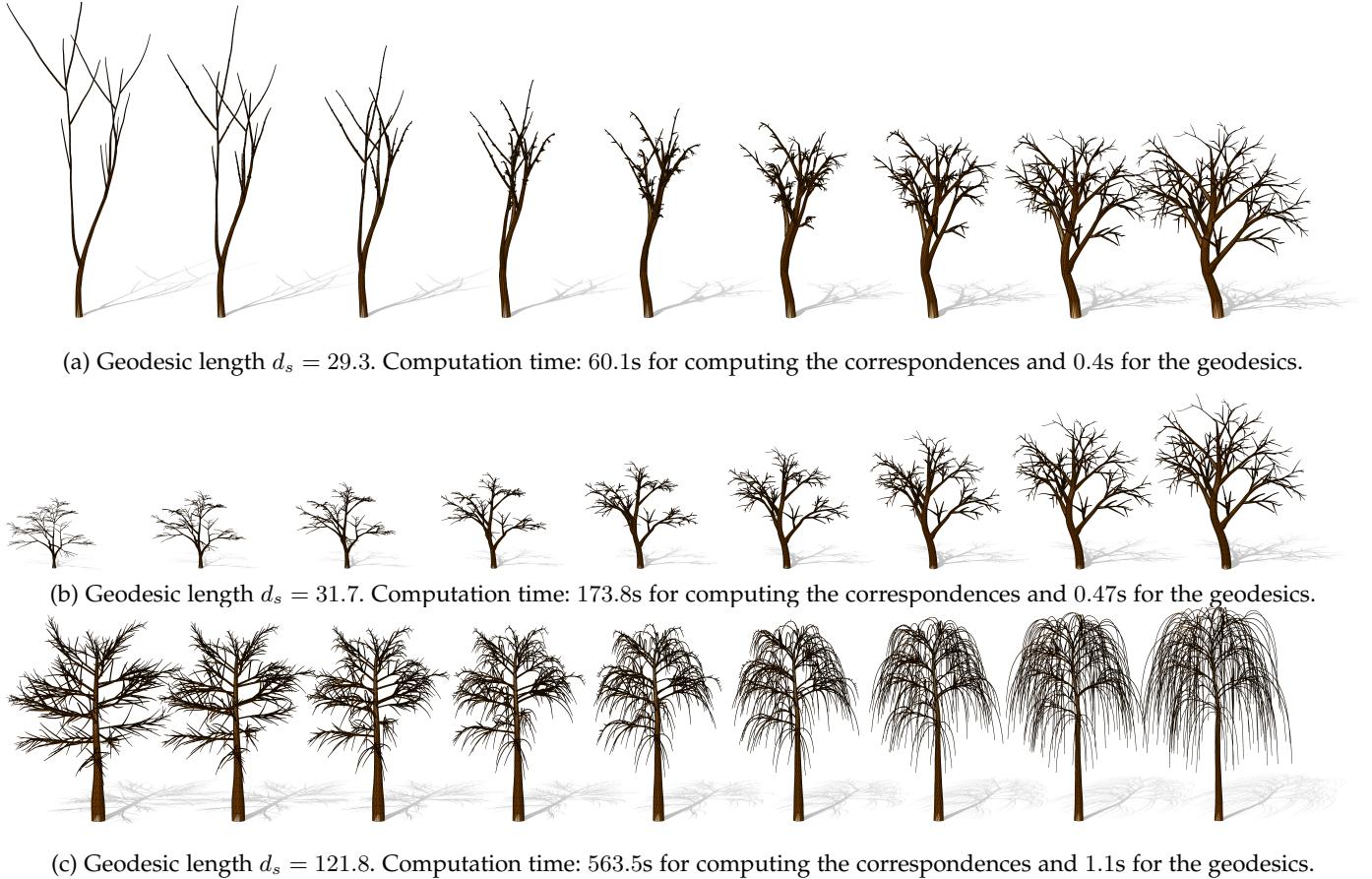
Fig. 3: Geodesic deformations between the most left and the most right 3D botanical trees in each row. In this experiment, we use $\lambda_m = \lambda_s = \lambda_p = 1.0$. We refer the reader to Fig. 14 in the Supplementary Material, which shows the color-coded subtree-wise correspondences. The Supplementary Material also includes more examples.

of such shapes (Section 8.3), computing summary statistics such as means and modes of variation (Section 8.4), and finally synthesizing novel tree-shaped 3D objects (Section 8.5). We use two types of datasets, namely 3D botanical trees obtained from xfrog.net[2], and 3D neuronal structures from [20], [21], [22]. In both cases, the models have complex branching structures composed of multiple layers, unlike Duncan *et al.* [18], which use simple trees of level one. The models used in this section also exhibit large geometric and topological variability, making it challenging to find one-to-one correspondences between such tree-like 3D objects. In all our experiments, we assumed that the main branch (trunk) of the botanical trees is the one that is less bent compared to the other branches. In the case of neuronal structures, the longest branch is selected as the main one. Also, we omit the branch thickness when computing correspondences. It is, however, included when computing geodesics and summary statistics.

The Supplementary Material includes more results as well as videos of all the examples shown in this paper.

8.1 Correspondence and geodesics

Figs. 3 and 4 show examples of geodesic paths between 3D botanical trees and 3D neuronal structures, respectively.

2. https://www.xfrog.net/xfrog-software/

For each example, we show the geodesic between the leftmost (source) and the rightmost (target) trees. The botanical trees exhibit very complex 3D geometric and topological structures. Nevertheless, the approach can find one-to-one correspondences and a natural deformation between the source and target trees. The neuronal trees of Fig. 4 have a relatively simple structure since they are composed of a main branch and a number of lateral branches. They, however, have a very complex geometry since their branches significantly bend and stretch. Finally, Fig. 5 shows the geodesic, computed with the proposed approach, between a small tree and a large tree.

Fig. 6 compares the results of our approach with the approaches of Feragen *et al.* [14] and Wang *et al.* [16]. Conceptually, the approach of Feragen *et al.* [14], which automatically finds correspondences and geodesics, does not consider the thickness of the branches that compose the tree shapes. It is also computationally costly, especially for complex trees. Wang *et al.* [16] extended the approach by pre-computing the correspondences in an ad-hoc manner. In both methods, the nodes represent bifurcation points, and the edges represent the branch segments between the bifurcation points. Topological changes are modeled using edge collapse and node split operations. As such, the intermediate shapes along a geodesic exhibit significant shrinkage since the edge collapse operation reduces the size of the collapsed edge to
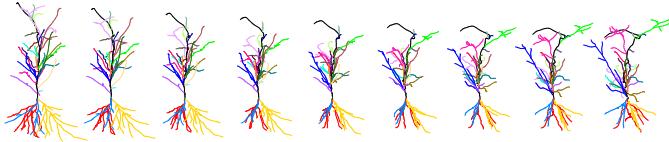
(a) Geodesic length $d_s = 156.7$. Computation time: 22.4s for computing the correspondences and 0.21s for the geodesics.
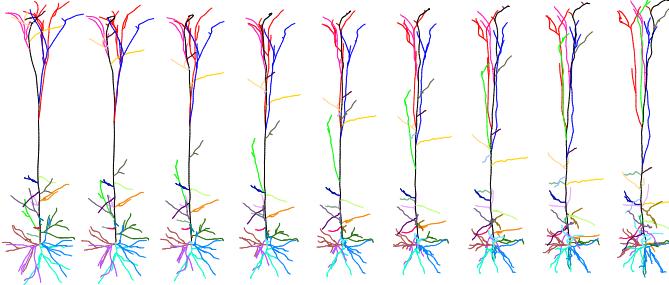
(b) Geodesic length $d_s = 165.2$. Computation time: 29.6s for computing the correspondences and 0.25s for the geodesics.
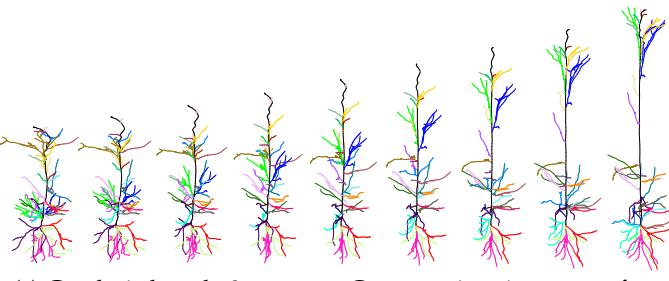
(c) Geodesic length $d_s = 181.8$. Computation time: 30.2s for computing the correspondences and 0.22s for the geodesics.

Fig. 4: Geodesic deformations between the most left and the most right 3D neuronal trees in each row. In this experiment, we use $\lambda_m = 0.2, \lambda_s = 1.0, \lambda_p = 0.2$. In each row, we show the geodesic between the most left and the most right trees as well as the color-coded subtree-wise correspondences.

Fig. 5: A geodesic path between a small tree and a large botanical trees. We refer the reader to Fig. 15 in the Supplementary Material, which shows the color-coded subtree-wise correspondences

zero, which shrinks the branch and subsequently the entire tree; see the bottom two rows in Fig. 6. This will affect downstream applications such as average tree computation and tree synthesis from a collection of 3D tree models. The approach proposed in this paper solves this problem by modeling topological changes as sliding branches. It also uses a full elastic metric, which explicitly quantifies the bending and stretching of the branches. Subsequently, the geodesics look more natural compared to the state-of-the-art; see the top row in Fig. 6.

Computation time. Our framework is implemented using

Matlab and runs entirely on CPU, configured with 2.4GHz Intel Core i5, 8GB RAM. The most time-consuming part is the correspondence computation process, which takes 60.1 to 4923.1 seconds for the botanical trees and 22.4 to 30.2 seconds for the neuronal trees. The subsequent geodesic computation takes significantly less time (0.4 to 1.9 seconds for the botanical trees and 0.2 seconds for the neuronal trees). We refer the reader to the captions of the figures for the computation time of each example.

8.2 Ablation study

Effect of the weights $\lambda_m, \lambda_s,$ and λ_p. These three parameters control the importance of each of the three terms in Eqn. (15). In the experiments of Fig. 3, these parameters were manually set to $\lambda_m = \lambda_s = \lambda_p = 1.0$ while in Fig. 4, they were set to $\lambda_m = 0.2, \lambda_s = 1.0, \lambda_p = 0.2$. Fig. 7 shows how the correspondences and geodesics vary while varying these parameters. For the first example in Fig. 7 with $\lambda_m = 1.0, \lambda_s = 10^{-4}, \lambda_p = 1.0$, the approach favors creating new virtual subtrees (see the subtree indicated by an arrow) rather than sliding the existing ones. We observe the opposite behavior in the second example when we set $\lambda_m = 1.0, \lambda_s = 1.0, \lambda_p = 10^{-4}$. This is reasonable since in the first example, the last term in Eqn. 12, which measures the distance between bifurcation points, is significantly higher than the second term, which measures shape dissimilarity between two subtrees.

Effect of the matching step. A core contribution of our framework is the shape matching between complex tree pairs. In this section, we investigate the effect of this step on the quality of the computed geodesics. Figs. 8 and 9 compare the geodesics obtained using our approach with and without the shape matching component. We can see that the geodesics generated by the framework with shape matching look more natural than those without shape matching. These results demonstrate the contribution of shape matching to the quality of the geodesic generation. The framework enables the least morphing cost between tree pairs by shape matching, thereby resulting in more natural and smoother morphing sequences.

8.3 Reflection symmetry analysis and symmetrization

Symmetry is an important feature of artificial and biological objects and its analysis can benefit many applications. Existing symmetry analysis techniques are mainly based on feature detection and matching; see, for example, the survey by Mitra *et al.* [44]. The proposed framework, which provides a proper metric and a mechanism for computing geodesics, can be used to analyze the reflection symmetry and to symmetrize tree-shaped 3D objects without extracting and computing shape descriptors.

To analyze the level of asymmetry of a given tree-shaped object $\boldsymbol{\beta}$ using our framework, we first obtain its reflection across an arbitrary plane Δ. Let $v \in \mathbb{R}$ be a normal vector to this plane. Then, the reflection of $\boldsymbol{\beta}$ with respect to Δ is given by

$$\tilde{\boldsymbol{\beta}} = \left(I - 2\frac{vv^\top}{v^\top v}\right)\boldsymbol{\beta}. \tag{22}$$
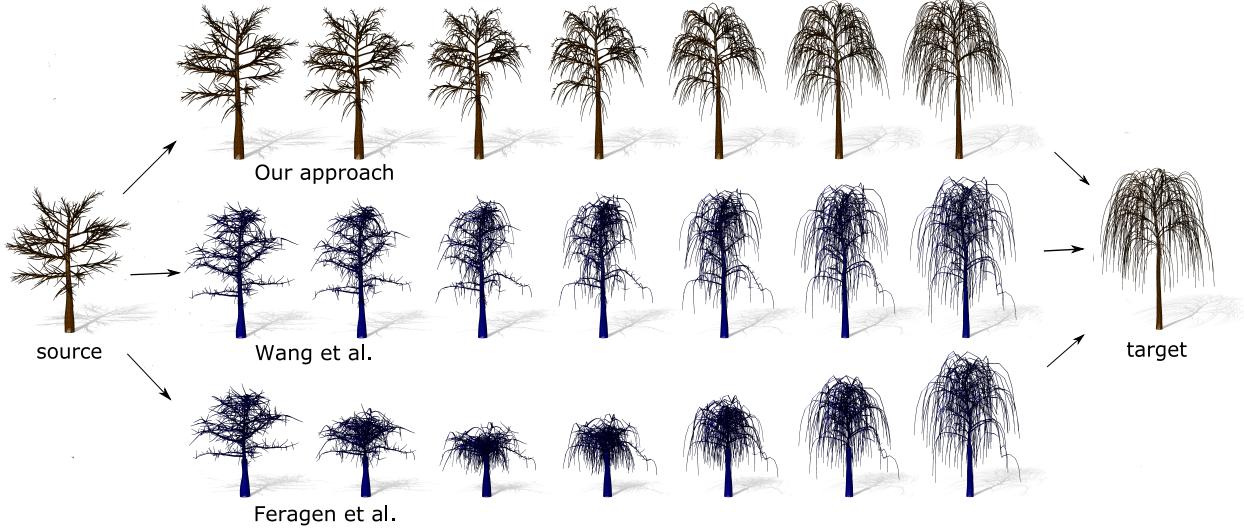
Fig. 6: Comparison of the quality of the geodesics obtained using our approach, the approach of Feragen *et al.* [14], and the approach of Wang *et al.* [16]. The Supplementary Material includes more comparison examples.


(a) $\lambda_m = 1.0, \lambda_s = 10^{-4}, \lambda_p = 1.0$.

(b) $\lambda_m = 1.0, \lambda_s = 1.0, \lambda_p = 10^{-4}$.
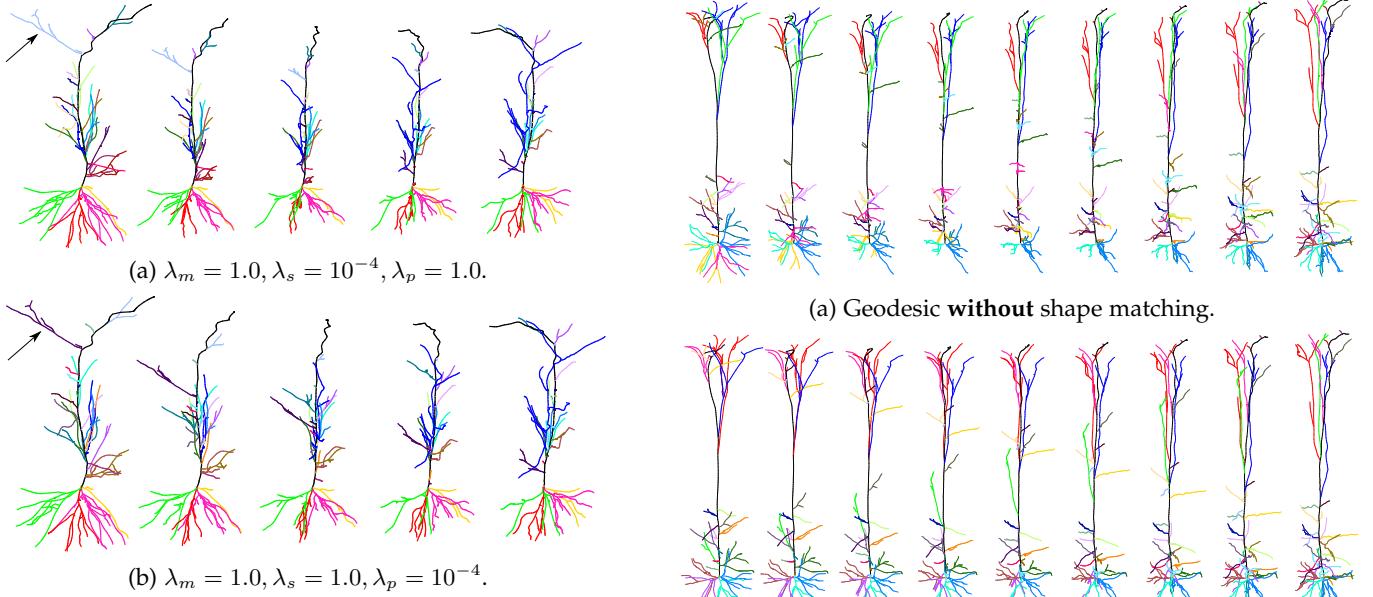
Fig. 7: The influence of weights λ_m, λ_s, and λ_p of Eqn. 12 on the geodesics computed with our approach. The colors indicate subtree-wise correspondences.

(a) Geodesic **without** shape matching.

(b) Geodesic **with** shape matching.

Fig. 8: Ablation study on neuronal tree geodesics. (a) Geodesic generated without graph matching, (b) geodesic generated with graph matching. In each row, we show the geodesic between the most left and the most right trees as well as the color-coded subtree-wise correspondences.

Here, I is the 3×3 identity matrix and $v^{\top}$ refers to the transpose of the vector v. Next, we use the approach described in this paper to compute a geodesic path, α, between β and $\tilde{\beta}$. α provides valuable information about the symmetry of β. First, its length gives a proper measure of the asymmetry of α. Second, the halfway point along this geodesic, *i.e.*, $\alpha(0.5)$, is symmetric. Moreover, amongst all symmetric trees, $\alpha(0.5)$ is the nearest to β. Thus, the process of computing a geodesic between a tree-shaped object and its reflection is equivalent to symmetrizing the object.

Fig. 10 shows two examples of complex botanical trees, and Fig. 11 shows one example of neuronal structures sym-

metrized using the proposed approach. Observe that the midpoint of each geodesic is symmetric; thus, the geodesic path provides a natural symmetrization of the given tree-shaped 3D object.

8.4 Summary statistics

Fig. 12 shows one example of mean shapes computed from three 3D neuronal trees. The Supplementary Material

(a) Geodesic **without** shape matching.

(b) Geodesic **with** shape matching.
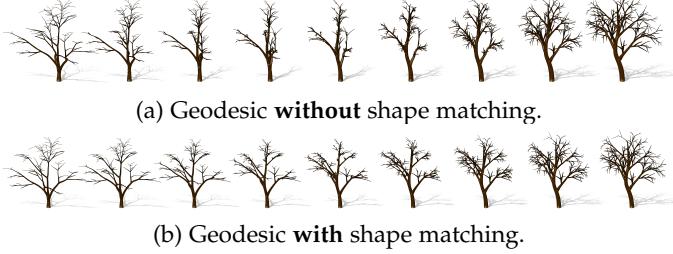
Fig. 9: Ablation study on botanical tree geodesics. (a) Geodesic generated without graph matching, (b) geodesic generated with graph matching. We refer the reader to Fig. 16 in the Supplementary Material, which shows the color-coded subtree-wise correspondences.


(a) Geodesic length $d_s = 4.7$.

(b) Geodesic length $d_s = 26.6$.
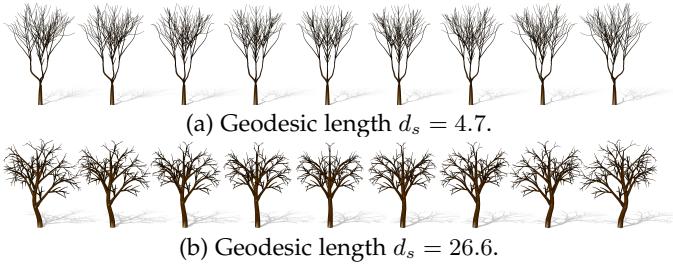
Fig. 10: Analysis of the symmetry and symmetrization of 3D botanical trees. The Supplementary Material includes more examples. We refer the reader to Fig. 17 in the Supplementary Material, which shows the color-coded subtree-wise correspondences.

provides more examples of mean shapes of 3D neuronal trees and for 3D botanical trees. From these results, we can see that the mean shapes computed with the proposed framework capture the overall shape information of the input samples.

Next, we perform a covariance analysis of a collection of 36 models of botanical trees. Fig. 13 shows the first three leading modes of variation (one per row). The middle shape in each row (at zero standard deviation) is the mean shape of the input collection. We also perform a similar experiment on 51 neuronal structures; see Fig. 9 in the Supplementary Material. These two experiments show that the leading modes of variation capture the main geometric and structural variation of the input 3D tree shape collections.

8.5 3D Tree-shape generation

Finally, once a statistical model is fit to a collection of tree-shaped 3D objects, one can synthesize new 3D shape instances via random sampling from the statistical model. Fig. 14 shows examples of randomly synthesized samples. These tree shapes have been produced without any professional knowledge. We can clearly see that they share some shape similarities with the input trees (see Fig. 7 in the Supplementary Material), but are not identical.

8.6 Quantitative analysis on model performance

We quantitatively evaluate the performance of the proposed framework and compare it to Wang *et al.* [16] using the


Geodesic length $d_s = 639.7$.
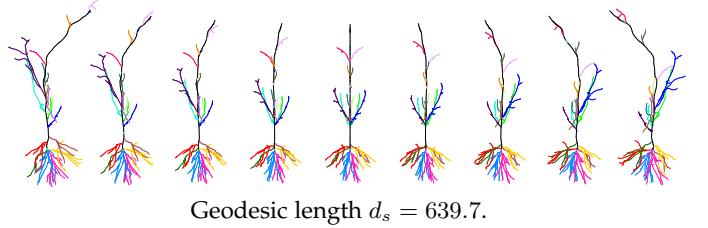
Fig. 11: Symmetry and symmetrization of 3D neuronal structures. We show the geodesic between the most left and the most right trees as well as the color-coded subtree-wise correspondences. The Supplementary Material includes more examples.

(a) mean shape of three neuronal trees.
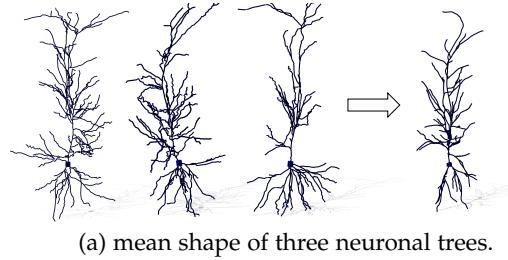
Fig. 12: Mean shape of 3D neuronal trees. The Supplementary Material includes more examples.

Minimum Description Length (MDL) criteria. Specifically, we use a botanical tree collection composed of 6 examples and compute the eigenvectors $\{\lambda_i\}_{i=1}^m$ and the eigenvalues $\{\Lambda_i\}_{i=1}^m$ of their covariance matrix K in their shape space. We record the captured information of the top-k eigenvalues by computing the ratio $\frac{\sum_{i=1}^k \lambda_i}{\sum_{i=1}^m \lambda_i}, k \leq m$. A good model is the one that captures maximum information with a minimum number k of eigenvalues. Table 1 reports the ratios for top-1 to top-5 eigenvalues and their corresponding shape space dimensions. As one can see, using the same number of eigenvalues and eigenvectors, our framework captures more information than the approach of Wang *et al.* [16], even when using the same number of eigenvalues/eigenvectors.

9 CONCLUSION

In this paper, we developed a comprehensive framework for quantifying the shape of biological 3D objects that deform both in geometry and topology. These objects are characterized by variable branching structures, sizes, and shapes. This framework provides tools for quantifying shape differences, computing geodesic deformations, and modeling

TABLE 1: Comparison of the performance of the proposed framework with Wang *et al.* [16]. We report the amount of information represented as a function of the number of eigenvalues/eigenvectors.

|	Number of eigenvalues					Space dim.
	1	2	3	4	5	m
Wang *et al.* [16]	0.54	0.79	0.92	0.97	**1.00**	2541
Our method	**0.60**	**0.81**	**0.94**	**0.99**	1.00	**4910**
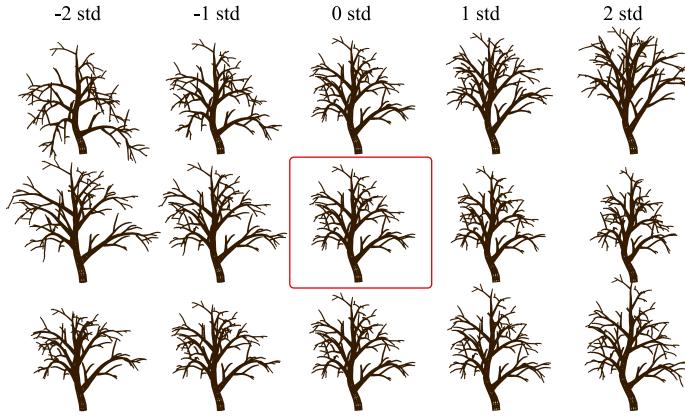
Fig. 13: Mean tree (highlighted in the middle) and the first three principal modes of variation (one per row) for 36 botanical trees. The input botanical trees are shown in the supplementary material.
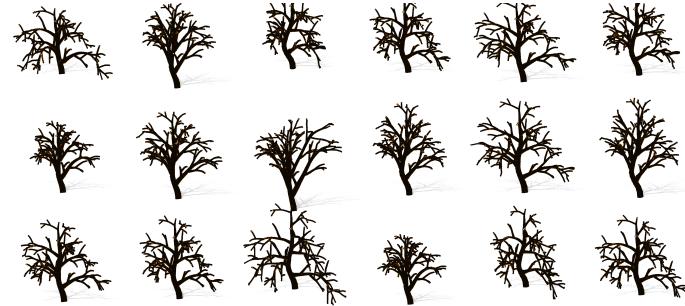
Fig. 14: Randomly synthesized botanical trees. The Supplementary Material includes more examples.

shape variability. A key idea here is to match common substructures across objects to result in more natural deformations and shape summaries. The resulting framework provides a powerful setting for quantifying complex structural variability in tree-like 3D objects such as neuronal structures and botanical tree models.

Although effective as evidenced by the results presented in this paper, there are several improvements that can be considered for future work. First, the geodesic computation algorithm is based on a gradient-descent approach and thus converges to a local minimum. In practice, many heuristics can be used to approach the global minimum, *e.g.*, by running the algorithm multiple times, each time with a random initialization, and then choosing the optimal solution. While effective, this approach can result in an increase in computation time. Thus, we plan in the future to explore better optimization approaches. Second, the geodesic quality depends on the weights of the three terms of the distance metric defined in Eqn. 12. In this work, we manually set those weights, while in practice they depend on specific medical knowledge, *e.g.*, plant biology and medicine. Thus, one can consider learning these parameters directly from data labeled with domain-specific knowledge. Third, currently, we have only considered the synthesis by random sampling and not taking into account biological factors such as light intensity, aging processes, disease, etc., which affect

the 3D shape of tree-like structures such as botanical trees and neurons. One potential direction for future work is to incorporate these factors into the modeling process, *e.g.*, via regression. Fourth, we plan to explore in the future how to model deformations and growth patterns of tree-shaped 3D objects as trajectories in the proposed SRVFT space, similar to [45]. Fifth, we have not considered the functional correspondence and functionality when we align two trees in this paper. So in the future, we plan to incorporate the part functionality to achieve function-consistent geodesic. Finally, the proposed framework can be used to enrich data collections, and thus can benefit data-driven applications such as deep learning-based 3D reconstruction [41].

REFERENCES

[1] X. Jiang, S. Shen, C. R. Cadwell, P. Berens, F. Sinz, A. S. Ecker, S. Patel, and A. S. Tolias, "Principles of connectivity among morphologically defined cell types in adult neocortex," *Science*, vol. 350, no. 6264, p. aac9462, 2015.

[2] N. Spruston, "Pyramidal neurons: dendritic structure and synaptic integration," *Nature Reviews Neuroscience*, vol. 9, no. 3, pp. 206–221, 2008.

[3] C. Lin, Y. Huang, T. Quan, and Y. Zhang, "Modelling brain-wide neuronal morphology via rooted cayley trees," *Scientific reports*, vol. 8, no. 1, pp. 1–10, 2018.

[4] D. Kabaso, P. J. Coskren, B. I. Henry, P. R. Hof, and S. L. Wearne, "The electrotonic structure of pyramidal neurons contributing to prefrontal cortical circuits in macaque monkeys is significantly altered in aging," *Cerebral Cortex*, vol. 19, no. 10, pp. 2248–2268, 2009.

[5] V. Chan-Palay and E. Asan, "Alterations in catecholamine neurons of the locus coeruleus in senile dementia of the alzheimer type and in parkinson's disease with and without dementia and depression," *Journal of comparative neurology*, vol. 287, no. 3, pp. 373–392, 1989.

[6] M. J. West, P. D. Coleman, D. G. Flood, and J. C. Troncoso, "Differences in the pattern of hippocampal neuronal loss in normal ageing and alzheimer's disease," *The Lancet*, vol. 344, no. 8925, pp. 769–772, 1994.

[7] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*.   ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[8] S. Kurtek, A. Srivastava, E. Klassen, and H. Laga, "Landmark-guided elastic shape analysis of spherically-parameterized surfaces," in *Computer graphics forum*, vol. 32, no. 2pt4.   Wiley Online Library, 2013, pp. 429–438.

[9] H. Laga, S. Kurtek, A. Srivastava, and S. J. Miklavcic, "Landmark-free statistical analysis of the shape of plant leaves," *Journal of theoretical biology*, vol. 363, pp. 41–52, 2014.

[10] H. Laga, Q. Xie, I. H. Jermyn, A. Srivastava *et al.*, "Numerical inversion of srnf maps for elastic shape analysis of genus-zero surfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2451–2464, 2017.

[11] I. H. Jermyn, S. Kurtek, H. Laga, and A. Srivastava, "Elastic shape analysis of three-dimensional objects," *Synthesis Lectures on Computer Vision*, vol. 12, no. 1, pp. 1–185, 2017.

[12] A. Feragen, F. Lauze, P. Lo, M. de Bruijne, and M. Nielsen, "Geometries on spaces of treelike shapes," in *Asian Conference on Computer Vision*. Springer, 2010, pp. 160–173.

[13] A. Feragen, S. Hauberg, M. Nielsen, and F. Lauze, "Means in spaces of tree-like shapes," in *Computer Vision (ICCV), 2011 IEEE International Conference on*. IEEE, 2011, pp. 736–746.

[14] A. Feragen, P. Lo, M. de Bruijne, M. Nielsen, and F. Lauze, "Toward a theory of statistical tree-shape analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 2008–2021, 2013.

[15] A. Feragen, M. Owen, J. Petersen, M. M. Wille, L. H. Thomsen, A. Dirksen, and M. de Bruijne, "Tree-space statistics and approximations for large-scale analysis of anatomical trees," in *International Conference on Information Processing in Medical Imaging*. Springer, 2013, pp. 74–85.

[16] G. Wang, H. Laga, N. Xie, J. Jia, and H. Tabia, "The shape space of 3d botanical tree models," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 1, p. 7, 2018.

[17] G. Wang, H. Laga, J. Jia, N. Xie, and H. Tabia, "Statistical modeling of the 3d geometry and topology of botanical trees," in *Computer Graphics Forum*, vol. 37, no. 5. Wiley Online Library, 2018, pp. 185–198.

[18] A. Duncan, E. Klassen, A. Srivastava *et al.*, "Statistical shape analysis of simplified neuronal trees," *The Annals of Applied Statistics*, vol. 12, no. 3, pp. 1385–1421, 2018.

[19] A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, "Shape analysis of elastic curves in euclidean spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1415–1428, 2011.

[20] G. A. Ascoli, D. E. Donohue, and M. Halavi, "Neuromorpho. org: a central resource for neuronal morphologies," *Journal of Neuroscience*, vol. 27, no. 35, pp. 9247–9251, 2007.

[21] L. Suo, H. Lu, G. Ying, M. R. Capecchi, and Q. Wu, "Protocadherin clusters and cell adhesion kinase regulate dendrite complexity through rho gtpase," *Journal of molecular cell biology*, vol. 4, no. 6, pp. 362–376, 2012.

[22] J.-R. Chen, B.-N. Wang, G.-F. Tseng, Y.-J. Wang, Y.-S. Huang, and T.-J. Wang, "Morphological changes of cortical pyramidal neurons in hepatic encephalopathy," *BMC neuroscience*, vol. 15, no. 1, pp. 1–12, 2014.

[23] D. G. Kendall, "A survey of the statistical theory of shape," *Statistical Science*, vol. 4, no. 2, pp. 87–99, 1989.

[24] I. Dryden and K. Mardia, *Statistical Shape Analysis*. John Wiley & Son, 1998.

[25] B. Allen, B. Curless, B. Curless, and Z. Popović, "The space of human body shapes: reconstruction and parameterization from range scans," in *ACM transactions on graphics (TOG)*, vol. 22, no. 3. ACM, 2003, pp. 587–594.

[26] D. G. Kendall, D. Barden, T. K. Carne, and H. Le, *Shape and shape theory*. John Wiley & Sons, 2009.

[27] S. Bouix, J. Pruessner, D. Collins, and K. Siddiqi, "Hippocampal shape analysis using medial surfaces," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2001, vol. 2208, pp. 33–40.

[28] K. Gorczowski, M. Styner, J. Y. Jeong, J. S. Marron, J. Piven, H. C. Hazlett, S. M. Pizer, and G. Gerig, "Multi-object analysis of volume, pose, and shape using statistical discrimination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 4, pp. 652–61, 2010.

[29] S. Osher and R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces (Applied Mathematical Sciences)*, 2003rd ed., 2002.

[30] R. Davies, C. Twining, T. Cootes, and C. Taylor, "Building 3-D statistical shape models by direct optimization," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 4, pp. 961–981, April 2010.

[31] M. Kilian, N. J. Mitra, and H. Pottmann, "Geometric modeling in shape space," in *ACM SIGGRAPH*, 2007.

[32] B. Heeren, M. Rumpf, M. Wardetzky, and B. Wirth, "Time-discrete geodesics in the space of shells," *CGF*, vol. 31, no. 5, pp. 1755–1764, 2012.

[33] C. Zhang, B. Heeren, M. Rumpf, and W. A. Smith, "Shell PCA: statistical shape modelling in shell space," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1671–1679.

[34] O. Van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A Survey on Shape Correspondence," *CGF*, vol. 30, no. 6, pp. 1681–1707, 2011.

[35] H. Laga, Y. Guo, H. Tabia, R. B. Fisher, and M. Bennamoun, *3D Shape analysis: fundamentals, theory, and applications*. John Wiley & Sons, 2018.

[36] S. Kurtek, E. Klassen, J. C. Gore, Z. Ding, and A. Srivastava, "Elastic geodesic paths in shape space of parameterized surfaces," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1717–1730, 2011.

[37] I. H. Jermyn, S. Kurtek, E. Klassen, and A. Srivastava, "Elastic shape matching of parameterized surfaces using square root normal fields," in *European conference on computer vision*. Springer, 2012, pp. 804–817.

[38] L. J. Billera, S. P. Holmes, and K. Vogtmann, "Geometry of the space of phylogenetic trees," *Advances in Applied Mathematics*, vol. 27, no. 4, pp. 733–767, 2001.

[39] M. Owen and J. S. Provan, "A fast algorithm for computing geodesic distances in tree space," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 8, no. 1, pp. 2–13, 2011.

[40] B. Aydın, G. Pataki, H. Wang, E. Bullitt, J. S. Marron *et al.*, "A principal component analysis for trees," *The Annals of Applied Statistics*, vol. 3, no. 4, pp. 1597–1615, 2009.

[41] X.-F. Han, H. Laga, and M. Bennamoun, "Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 5, pp. 1578–1604, 2019.

[42] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[43] R. Jonker and T. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," in *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*. Springer, 1988, pp. 622–622.

[44] N. J. Mitra, M. Pauly, M. Wand, and D. Ceylan, "Symmetry in 3d geometry: Extraction and applications," in *Computer Graphics Forum*, vol. 32, no. 6, 2013, pp. 1–23.

[45] H. Laga, M. Padilla, I. H. Jermyn, S. Kurtek, M. Bennamoun, and A. Srivastava, "4d atlas: Statistical analysis of the spatio-temporal variability in longitudinal 3d shape data," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, pp. 1335–1352, 2023.
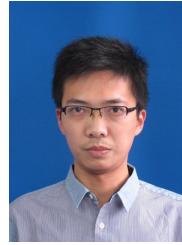
Guan Wang is currently an Associate Professor at Yangtze Delta Region Institute (Huzhou), University of Electronic Science and Technology of China (UESTC). He obtained his Ph.D degree in Software Engineering from Tongji University in 2019. After being a postdoc at Technical University of Denmark for 2 years, he joined UESTC in 2022. His research interests lie primarily in the fields of computer graphics and deep learning, especially in 3D tree modeling and graph neural networks.

Hamid Laga is a Professor at Murdoch University (Australia). His research interests span various fields of machine learning, computer vision, computer graphics, and pattern recognition, with a special focus on the 3D reconstruction, modeling, and analysis of static and deformable 3D objects, and on machine learning for agriculture and health. He is the recipient of the Best Paper Awards at SGP2017, DICTA2012, and SMI2006.

Anuj Srivastava is a Professor in the Department of Statistics and a Distinguished Research Professor at the Florida State University. He obtained his Ph.D. from Washington University in St. Louis in 1996. After being a postdoc at Brown University for one year, he joined the faculty of FSU in 1997. His research interests include statistical analysis on nonlinear manifolds, statistical computer vision, and functional and shape analysis. He has held several visiting positions at European universities, including INRIA, France, the University of Lille, France, and Durham University, UK. He has graduated 30+ Ph.D. students with placements in academia, industry, and government labs. He has co-authored more than 230 papers in peer-reviewed journals and top-tier conferences and several books, including the 2016 Springer textbook on "Functional and Shape Data Analysis." He is a fellow of IAPR, IEEE, ASA, and AAAS.