

STATISTICAL SHAPE ANALYSIS OF SIMPLIFIED NEURONAL TREES

BY ADAM DUNCAN, ERIC KLASSEN AND ANUJ SRIVASTAVA

Florida State University

Neuron morphology plays a central role in characterizing cognitive health and functionality of brain structures. The problem of quantifying neuron shapes and capturing statistical variability of shapes is difficult because neurons differ both in geometry and in topology. This paper develops a mathematical representation of neuronal trees, restricting to the trees that consist of: (1) a main branch viewed as a parameterized curve in \mathbb{R}^3 , and (2) some number of secondary branches—also parameterized curves in \mathbb{R}^3 —which emanate from the main branch at arbitrary points. It imposes a metric on the representation space, in order to compare neuronal shapes, and to obtain optimal deformations (geodesics) across arbitrary trees. The key idea is to impose certain equivalence relations that allow trees with different geometries and topologies to be compared efficiently. The combinatorial problem of matching side branches across trees is reduced to a linear assignment with well-known efficient solutions. This framework is then applied to comparing, clustering, and classifying neurons using fully automated algorithms. The framework is illustrated on three datasets of neuron reconstructions, specifically showing geodesics paths and cross-validated classification between experimental groups.

1. Introduction.

1.1. *Motivation and problem statement.* The human brain is a complex, dynamic, and multifaceted system that provides challenges at multiple functional and observation scales to researchers. Among different scientific objectives associated with current brain research, there is a great deal of interest in characterizing anatomical parts in terms of structures, functions, and their joint roles in cognitive processes. Since neurons are the basic units of our central nervous system, neuronal morphologies are key to understanding the progression of a pathology or to identify therapeutic targets. For example, the shapes of axons directly impact the number of contacts related neurons can have and are, therefore, highly correlated to the neuron network connectivity and signaling pathways [Cuntz et al. (2008), Hirokawa, Niwa and Tanaka (2010)]. Alterations in neuron morphology have been reported not only in normal aging [Kabaso et al. (2009)], but more importantly

Received August 2016; revised October 2017.

Key words and phrases. Neuron morphology, elastic shape analysis, tree registration, neuron deformation, tree geodesics.

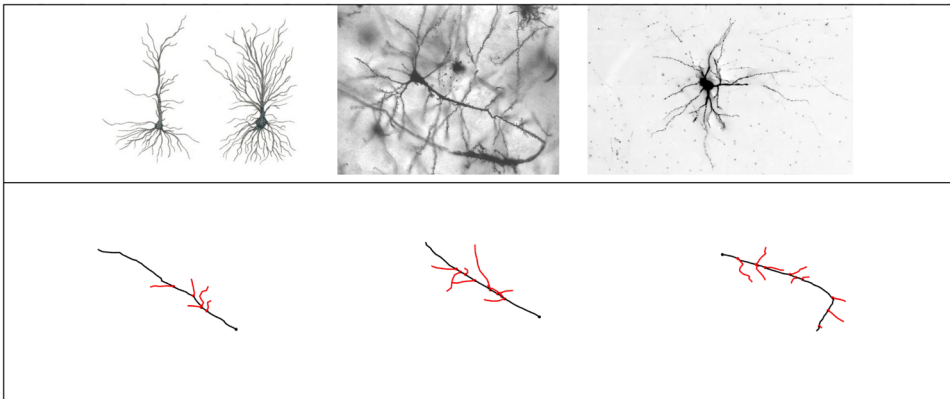


FIG. 1. *Top row: Some microscopy images of neurons illustrating the tree structures. Bottom row: Examples of simple structures taken from NeuroMorpho.org database (in the Wu dataset described in Section 3). These neurons differ in number and locations of the axon/dendrite branches as well as the geometries of those branches.*

in patients with senile dementia [Chan-Palay and Asan (1989)], Alzheimer's disease [Coleman and Flood (1987), West et al. (1994), Whitehouse et al. (1982)], and Fragile X syndrome [Bassell and Warren (2008)]. Therefore, a *comprehensive geometric and statistical morphological analysis* of neurons is crucial in understanding brain functionality. While the publicly available databases for studying neuron morphologies are growing rapidly, the techniques for analysis are lagging far behind, as noted in Ledderose et al. (2014). Recent years have seen many advances in imaging and extraction of 3D structures of neurons [Andersson-Engels et al. (1997), Ntziachristos (2006)], including software packages such as NeuroLucida and Neuromantic. The literature also contains methods and softwares for extracting neuronal networks from image data [see, e.g., the Diadem challenge (<http://diademchallenge.org/>)]. The top row of Fig. 1 shows some images of neurons, and the bottom row shows examples of relatively simple neurons taken from NeuroMorpho.org database. In this paper, we do not focus on the extraction problem, but instead focus on neuron morphology.

A major focus in any biological study is the comparison of populations. For better understanding and modeling of genetic differences between organisms, a widespread technique is gene knockout: an experimentally controlled mutation in which a single gene is made inoperative. This causes some change in the affected organism, such as the further initiation of some disease. Comparing the wild type population to the mutant one allows us to develop an understanding of the effect of the pathology on the tissues and/or on their evolution [Bassell and Warren (2008), Engle (2008), Medioni et al. (2014), Wu et al. (2012)]. Although the current statistical tests or classical metrics permit us to assess the existence of a significant difference between any two populations, they do not provide a structurally com-

plete description of this difference. In this context, the challenge is to construct proper metrics that are both effective and biologically interpretable.

This motivates the following goals:

1. *Shape Metric*: Given any two neurons, or rather their Euclidean coordinates, we want to quantify differences in their shapes. This requires defining an appropriate shape metric for comparing these structures.
2. *Cluster and Classify*: Using this shape metric, we want to cluster and classify neurons, and validate this classification with some covariates such as mutation, gender, or age.

The problem of comparing neuron structures is challenging because they exhibit variability in: (1) *Geometry*, in terms of shapes of individual branches of axons and dendrites, and (2) *Topology*, in terms of the structural relationship between those branches. One consequence of variable topology is that it makes it difficult to register parts across neurons. Registration is the process of finding an optimal (in a certain well-defined mathematical way) correspondence between parts—points, curves, branches, etc.—across tree-like structures of neurons. It is a difficult problem to solve when the objects being compared have different topologies and geometries. For instance, how should one match up branches across two trees where one tree has three branches and the other has ten?

1.2. *Current techniques*. Although there is a significant literature on neuron morphology, we naturally focus on techniques grounded in mathematical principles. The relevant works can be grouped in three broad categories:

1. *Feature Extraction and Analysis*: A majority of work in neuron morphology literature is based on extracting some *morphological features* of interests and then analyzing them quantitatively using Euclidean metrics. [See Halavi et al. (2012) for a broad review of these methods.] For example, Ledderose et al. (2014) use statistical distributions of basic morphological parameters such as branch length, tortuosity, branch's genealogy, and bifurcation angles. As another example, Neurolucida, one of the most commonly used software for structural analysis, extracts a slightly different set of features: number of dendrites, axons, nodes, synapses, and spines; the length, width, and volume of dendrites and axons; the area and volume of the soma. These extractions represent only partial information about neurons and often do not provide the full picture. In contrast, our interest is in a representation of axon and dendrite branching structures which allows us to compare those structures directly.

2. *Methods Using Tree Topology Alone*: Outside of neuromorphology, many researchers investigate the space of tree structures using topological approaches that ignore or simplify the geometry (shapes) of the branches and focus on comparing trees according to their branching patterns. In contrast to the methods based on feature extraction, these approaches compare entire trees to each other according

to their overall structure. Past work on purely topological tree analysis includes the famous tree-edit distance (TED) [Heuman and Wittum \(2009\)](#), [Selkow \(1977\)](#), [Tai \(1979\)](#), [Zhang \(1996\)](#), and its more recent extensions by [Wang and Marron \(2007\)](#), [Aydin et al. \(2011, 2009\)](#). These methods treat tree structures as abstract binary trees and focus on the locations of branches in the structural hierarchy. A path through the space of trees consists of a sequence of insertions and deletions of branches without regard for the geometry (shape) of the branches. Shapes of branches have important implications in scientific applications, and cannot be simply discarded in structural comparisons. To take into account both topological and geometrical variability requires novel mathematical tools.

3. *Combined Geometric-Topological Approaches*: There are also methods based on geometric spaces of trees which incorporate variation in tree topology. [Billera, Holmes and Vogtmann \(2001\)](#) develop a geometric space of phylogenetic trees based on branch lengths as one-dimensional continuous edge attributes. Trees with the same topology lie in a section of Euclidean space where the branch lengths are coordinate values, and each possible tree topology is associated with a copy of this space. The overall space is formed by gluing together these spaces at their boundaries.

More recently, this has been extended to trees with vector-valued edge attributes in a series of papers that includes [Feragen, Lauze and Hauberg \(2011, 2013a, 2013b, 2015\)](#). The authors have used a distance called Quotient Euclidean Distance (QED) based on geometric comparison between individual branches which also allows branches to be inserted or removed, thus changing the tree topology. Each branch is associated with a structural position in tree *and* a set of features giving geometric information about the branch—typically a vector of landmark points. In the spirit of Kendall’s shape analysis, the method starts from a preshape space in which tree structure is fixed, and the distance is simply the norm of the vector of Euclidean distances between individual branches that have corresponding positions in the tree structure. Trees of different structures are combined into a larger shape space by introducing equivalence relations where branches may be reindexed in geometry-preserving ways. In the ultimate shape space, distances are minimal geodesic path lengths which may involve branch deformation, insertion, or deletion. Unlike with TED, the cost to insert or delete a branch has variable cost in QED, depending on the geometric qualities of the branches. Also unlike TED, there is a cost associated with matching branches to each other if they have different shapes.

As far as we know, this last method is the most well developed and sophisticated kind of geometric tree shape analysis to date. Unfortunately, it suffers from a few drawbacks:

1. The shape metric used to compare individual branches is somewhat simplistic. To represent open curves, QED uses a vector of landmark coordinates, and

compares them by Euclidean distance. This implicitly assumes a fixed registration between the landmarks of different branches.

2. It is computationally expensive. Under QED, computing the distance between two trees depends on finding an optimal matching between the edges of the two trees. According to Feragen (2012), this is a NP-complete problem. This applies not only to QED, but to nongeometric TED as well.

This is a fundamental challenge with any method that requires matching between tree structures. One way to address this is by using prior knowledge of the application domain to help choose the matching. For example, Feragen et al. (2013b) apply QED to airway trees, and some of the major branches have anatomical names. By starting from a partial labeling as determined by experts, they reduce the problem size and make it computationally tractable. Another way to deal with the problem is by discarding some parts of the tree structure to keep the problem size manageable, such as in Feragen et al. (2013a).

3. The correspondence between branches under QED is one-to-one. If a branch in one tree closely resembles the combined shape of a pair of branches in another tree, the framework has no way to account for this.

The method that we propose in this paper is mainly an improvement on the first issue. We use a framework based on elastic shape analysis of continuous curves, and the next section gives an overview of the previous work we draw from. Regarding the latter two issues, our proposed method makes a different set of trade-offs. We represent trees in a fundamentally different way which reduces the computational complexity. Unfortunately, we achieve this by limiting the kinds of tree topologies we consider, so QED is more general with regard to tree topology, as are purely topological methods. The different representation also changes the set of possible matchings between trees, and neither method is more general than the other with regard to matching.

1.3. *Our solution.* The framework we propose relates to shape analysis of objects under different mathematical representations and metrics. Statistical shape analysis includes a class of methods that deal with discrete sets of labeled points, or landmarks, and one uses geometries of such representation spaces to perform statistical procedures [Dryden and Mardia (1998), Kendall et al. (1999)]. A major limitation of this approach is that it assumes the registration of landmarks across shapes is provided with the data, which in practice, is often not possible.

Another class of shape analysis methods deal with continuous representations (\mathbb{R}^n -valued functions on some compact domains) and solve for registration during shape comparisons. Examples of shapes usually studied using these methods include parameterized curves and 2D surfaces. These methods rely on choosing a *shape metric* that is invariant to certain shape preserving transformations. For instance, in shape analysis of curves in Euclidean spaces, it is important to design a metric that is invariant to rigid motion, global scaling, and reparameterizations

of curves [Kurtek et al. (2012), Srivastava et al. (2011)]. This has led to creation of a new area of research where one designs invariant Riemannian metrics, termed *elastic metrics*, for comparing geometries of objects with the same topology. In this approach, one performs registration of points across shapes at the same time as computation of the shape metric, rather than as a preprocessing step. Consequently, there has been substantial amount of work in elastic shape analysis of curves in Euclidean spaces [Srivastava and Klassen (2016)].

This idea has also been extended to study *annotated* curves, where the analysis relies not only on the geometry of curves, but also on annotations defined as finite-dimensional functions defined along those curves [Liu, Srivastava and Klassen (2008)], with applications to protein structure analysis [Liu, Srivastava and Zhang (2011)]. Duncan et al. (2015) further extends the idea of annotated curves to infinite-dimensional attributes, where the annotation is itself another curve.

We propose a framework that can handle both geometrical and topological variability of neuronal trees, and can quantify differences between neuronal trees based on the shape, which includes information about the size, location, and number of branches. It modifies and extends the prior work on shape analysis of curves and annotated curves to include trees that consist of: (1) a main branch viewed as a parameterized curve in \mathbb{R}^3 , and (2) some number of secondary branches—also parameterized curves in \mathbb{R}^3 . These secondary branches are side branches of the main branch, and they begin at some point along the main branch curve. The framework imposes an elastic metric on the representation space and uses that to compare shapes of neuronal trees. In the process, it computes optimal registrations and deformations (geodesics) between any two trees. The key idea is to impose certain equivalence relations that allow trees with different geometries and topologies to be compared efficiently. The combinatorial problem of matching side branches across trees is reduced to a linear assignment with well-known efficient solutions. This framework is then applied to perform classification of neurons according to experimental group.

In the setup described above, we handle bifurcations in a completely different way than other works on tree spaces. In other tree space methods, such as QED, a bifurcation is a meeting point of three or more branches: a parent branch ends, and two or more child branches begin (parent-child). This is consistent with how trees are thought of as data structures, or as a type of graph where branches are edges, and bifurcations are vertices. In contrast, our method treats a bifurcation as a point where two branches meet: a side branch begins, and the main branch continues before and after the bifurcation (main-side). This is motivated by a type of branching in axons called *collateral formation*, which are side branches that sprout from somewhere along the shaft of a main axon branch. Figure 1 of Gibson and Ma (2011) illustrates collateral branching in contrast with other patterns of axon branching.



FIG. 2. An example pair of trees for which main-side is preferable to parent-child branching. (a) and (b) are two simple trees that have 0 and 1 bifurcations, respectively. (c) and (d) show two different ways of breaking up (b) into component branches, depending on how we think of the relationship between branches at the bifurcation. In (c), the parent branch ends and two child branches begin. In (d), the main branch continues through the bifurcation and one small child branch begins there. This main branch is identical to the sole branch of (a), and different from the branches of (c). A matching between the two trees is more satisfying if we can pair up the identical branches.

Previous work on tree spaces makes effective use of the parent-child concept of branching, but Figure 2 shows a simple example where the main-side concept is more appropriate. A more general framework might incorporate both concepts of branching, but we do not explore that here. One downside to the main-side approach is that it requires us to choose which branch we consider the main one. In Section 3, we discuss how we do this for datasets of simple neurons.

2. Mathematical representation and shape space metric. In this section, we describe mathematical representation and the chosen shape metric for comparing neuronal trees. Our framework treats branches as elastic curves and Section 2.1 gives a brief summary of the shape space of curves as it relates to this paper. Section 2.2 defines a tree preshape space where trees have the same number of side branches, and Section 2.4 extends this to a shape space of the more general case of having arbitrary number of branches.

2.1. *Elastic shape analysis of curves.* We use a version of the curve shape space described in papers such as Joshi et al. (2007), Kurtek et al. (2012), Mio and Srivastava (2004), Srivastava et al. (2011). Here, we only give an overview which includes definitions and key facts about this framework. For more details, see those papers.

Let \mathcal{AC} be the set of absolutely continuous parameterized curves from $[0, 1]$ to \mathbb{R}^3 . We endow \mathcal{AC} with a geometry given by an elastic Riemannian metric, which is a linear combination of two terms that measure bending and stretching of curves. Under an elastic metric, the distance between two curves can be thought of as a measure of how much smooth deformation it takes to transform one curve into the other. There is a convenient special case of such elastic metrics in which the space of curves can be flattened to infinite-dimensional Euclidean space by a transformation which we now define. Given $\beta \in \mathcal{AC}$, its *Square Root Velocity*

Function (SRVF) is defined as

$$(1) \quad q(t) = \begin{cases} \frac{\dot{\beta}(t)}{\sqrt{\|\dot{\beta}(t)\|}} & \text{if } \dot{\beta}(t) \text{ exists and is nonzero,} \\ 0 & \text{otherwise.} \end{cases}$$

The absolute continuity of β implies that q is square-integrable, so we will simply refer to the space of all such SRVFs as \mathbb{L}^2 , shorthand for $\mathbb{L}^2([0, 1], \mathbb{R}^3)$, the space of square-integrable functions from the unit interval to \mathbb{R}^3 . Given an SRVF, q , one can recover the original curve (up to translation) with the following integral:

$$(2) \quad \beta(t) - \beta(0) = \int_0^t q(r) \|q(r)\| dr.$$

The standard Euclidean inner product on \mathbb{L}^2 exactly corresponds to a Riemannian metric on \mathcal{AC} in the elastic family of metrics described above. Thus, geodesics in \mathcal{AC} correspond to straight lines in \mathbb{L}^2 and the elastic distance between two curves, $\beta_1, \beta_2 \in \mathcal{AC}$, can be computed by the Euclidean distance between their SRVFs, $q_1, q_2 \in \mathbb{L}^2$:

$$(3) \quad d_{\mathcal{AC}}(\beta_1, \beta_2) = \|q_1 - q_2\|_{\mathbb{L}^2}.$$

For shape analysis of curves, we use \mathbb{L}^2 as the preshape space, then mod out the shape-preserving group actions of reparameterization and rigid rotation to form the curve shape space. The reparameterizations come from the group of increasing, absolutely continuous bijections of the unit interval to itself:

$$(4) \quad \Gamma = \{\gamma : [0, 1] \rightarrow [0, 1] \mid \gamma \in \mathcal{AC}, \gamma(0) = 0, \gamma(1) = 1, \dot{\gamma} > 0 \text{ a.e.}\}.$$

This group acts by right composition of the original curve in \mathcal{AC} , so in the SRVF representation, the action is defined by $(q, \gamma) = \sqrt{\dot{\gamma}}(q \circ \gamma)$. The rotation action is left-multiplication by matrices in $\text{SO}(3)$ with all points, $\beta(t) \in \mathbb{R}^3$, along the curve, or equivalently left-multiplication with its SRVF representation. The two actions commute. For a given pair, $(\gamma, O) \in \Gamma \times \text{SO}(3)$, and SRVF, $q \in \mathbb{L}^2$, we can write the combined action $O(q, \gamma)$. The distance (3) is isometric with respect to this action, that is, for any $(\gamma, O) \in \Gamma \times \text{SO}(3)$:

$$(5) \quad \|q_1 - q_2\|_{\mathbb{L}^2} = \|O(q_1, \gamma) - O(q_2, \gamma)\|_{\mathbb{L}^2}.$$

For the shape space of curves alone, we take the quotient space of closures of orbits under this combined action:

$$[q] = \text{cl}(\{O(q, \gamma) \mid (O, \gamma) \in \Gamma \times \text{SO}(3)\}) \quad \text{and} \\ \mathcal{S} = \{[q] \mid q \in \mathbb{L}^2\}.$$

Elements in this quotient space (equivalence classes in the preshape space) are individual curve shapes, and the shape space inherits the standard distance pseudometric from the preshape space

$$(6) \quad d_S([q_1], [q_2]) = \inf_{\gamma, O \in \Gamma \times \text{SO}(n)} \|q_1 - O(q_2, \gamma)\|_{\mathbb{L}^2}.$$

The following lemma from [Lahiri, Robinson and Klassen \(2015\)](#) (Lemma 1 in that paper) establishes positivity for a slightly different quotient metric.

LEMMA 1. *Assume q_1 and q_2 are elements of $\mathbb{L}^2([0, 1], \mathbb{R}^N)$. Then $\inf_{\gamma \in \Gamma} \|q_1 - (q_2, \gamma)\|_{\mathbb{L}^2} = 0$ if and only if the orbits of q_1 and q_2 under reparameterization have the same closure in $\mathbb{L}^2([0, 1], \mathbb{R}^N)$.*

Another result that is useful for our purposes is Theorem 4 in the same paper, which establishes sufficient conditions for the existence of minimizers.

THEOREM 1. *Let $q, w \in \mathbb{L}^2([0, 1], \mathbb{R}^N)$ such that w is the SRVF of a piecewise linear curve. There exist some $\tilde{q} \in \text{cl}(q\Gamma)$ and $\tilde{w} \in \text{cl}(w\Gamma)$ such that $d(\tilde{q}, \tilde{w}) = \inf_{\gamma \in \Gamma} \|q - (w, \gamma)\|_{\mathbb{L}^2}$.*

Also, note that (6) is expressed in terms of the group action applied to only one of the curves q_2 , which is justified by the isometry stated in (5). Finding O, γ that minimize $d_{\mathbb{L}^2}$ is called curve alignment, or registration. Unfortunately, such a minimizing alignment may not exist, which is why we use closures of orbits to define the quotient space. In those cases, it is still, by definition, possible to find O, γ which make $d_{\mathbb{L}^2}$ as close to the infimum as possible. Framing the quotient distance this way sets up the alignment problem in a computationally convenient way that allows good approximate solutions.

[Lahiri, Robinson and Klassen \(2015\)](#) and [Bruveris \(2015\)](#) give similar constructions of curve shape space in which optimal alignments *do* generally exist. This is achieved by allowing reparameterizations to be non-decreasing, rather than strictly increasing. Consequently, the space of reparameterizations is a semigroup which includes noninvertible elements, and the alignment is done by simultaneously reparameterizing both curves. Such approaches are more exact, but less computationally tractable.

2.2. *Trees with the same number of side branches.* We consider simple trees which consist of a main branch and a finite (possibly empty) set of side branches that each begin at some arbitrary locations on the main branch. A simple tree of this type is written $\beta = (\beta_0, \{\beta_k\}_{k=1}^n)$. Each $\beta_k, k = 0, 1, \dots, n$ is an absolutely continuous curve, $\beta_k : [0, 1] \rightarrow \mathbb{R}^3$. The side branches are constrained to begin at some point on the interior of the main branch. That is, for each k , there is some

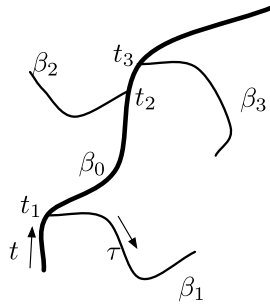


FIG. 3. Diagram of a simple tree with a main branch and two side branches.

$t_k \in (0, 1)$ such that $\beta_0(t_k) = \beta_k(0)$. A labeled diagram of this sort of tree is shown in Figure 3.

For the purpose of elastic shape analysis, each tree, β is represented by its square root velocity tree, \mathbf{q} , which consists of the collection of SRVFs of the curves in β , indexed in the same fashion. Since the SRVF representation is only unique up to translation, our representation should also keep information about the starting location for each side branch. A side branch is represented by the following ordered pair:

$$(7) \quad (q_k, s_k) \in \mathbb{L}^2([0, 1], \mathbb{R}^3) \times (0, 1),$$

where $s_k \in (0, 1)$ is the starting location expressed as the proportion of arc-length along the main branch

$$(8) \quad s_k = \frac{\int_0^{t_k} \|\dot{\beta}_0(t)\|_2 dt}{\int_0^1 \|\dot{\beta}_0(t)\|_2 dt} = \frac{\int_0^{t_k} (q_0(t))^2 dt}{\|q_0\|_{\mathbb{L}^2}^2}.$$

The main branch is also transformed to its SRVF, which we call q_0 . Our initial representation of the entire tree is this q_0 along with n side branches:

$$\mathbf{q} = (q_0, \{(q_k, s_k)\}_{k=1}^n).$$

For a given n , let \mathcal{P}_n be a product space in which \mathbf{q} resides. The q_k are all in \mathbb{L}^2 and the s_k are in $(0, 1)$, so the preshape space is

$$\mathcal{P}_n \equiv \mathbb{L}^2 \times (\mathbb{L}^2 \times (0, 1))^n.$$

Within this space, we define a distance in terms of distances in their component spaces. The curves, q_k , have \mathbb{L}^2 distance, and the branch positions, $s_k \in (0, 1)$, have a Euclidean distance. Given $\mathbf{q}^1, \mathbf{q}^2 \in \mathcal{P}_n$, we write each as $\mathbf{q}^i = (q_0^i, \{(q_k^i, s_k^i)\}_{k=1}^n)$ and define the square distance between them as a weighted sum of their component square distances:

$$(9) \quad d_n(\mathbf{q}^1, \mathbf{q}^2)^2 = \lambda_m \|q_0^1 - q_0^2\|^2 + \lambda_s \sum_{k=1}^n \|q_k^1 - q_k^2\|^2 + \lambda_p \sum_{k=1}^n (s_k^1 - s_k^2)^2.$$

The tuning parameters $\lambda = (\lambda_m, \lambda_s, \lambda_p)$ control the relative costs of deforming the main branch, deforming side branches, and moving the position of side branches, respectively. Since this distance is a weighted Euclidean norm of \mathbb{L}^2 distances and Euclidean distances, the combined space is flat. Thus, geodesics are just linearly interpolated paths. That is, each component is linearly interpolated: main branch, side branches, and side branch locations,

$$(10) \quad \alpha(r) = (1 - r)q^1 + rq^2, \quad 0 \leq r \leq 1.$$

Here, multiplication of a real number with a tree just means multiplication with each of its components, and addition of two trees is just addition of their corresponding components. Figure 4(b) illustrates an example geodesic path between two trees in \mathcal{P}_4 . In their initial representation, the two trees are oriented differently. To reach the second tree from the first, the path must shrink the features of the first tree while growing the features of the second. It passes through trees in \mathcal{P}_4 which do not much resemble the given trees.

2.3. *Removing rotation and reparameterization.* In the spirit of Srivastava et al. (2011), we can address this by modding out shape-preserving group actions. In each \mathcal{P}_n , we mod out the nuisance variables rotation and reparameterization. We apply rotation globally but reparameterize each branch separately, including the main branch. Let $\boldsymbol{\gamma} = (\gamma_0, \{\gamma_k\}_{k=1}^n) \in \Gamma^{n+1}$ be the ordered collection of reparameterizations to apply. Given $\boldsymbol{q} \in \mathcal{P}_n$ and the pair $(O, \boldsymbol{\gamma}) \in \text{SO}(3) \times \Gamma^{n+1}$, the group action is given by

$$(11) \quad O(\boldsymbol{q}, \boldsymbol{\gamma}) = (O(q_0, \gamma_0), \{O(q_k, \gamma_k), s_k\}_{k=1}^n).$$

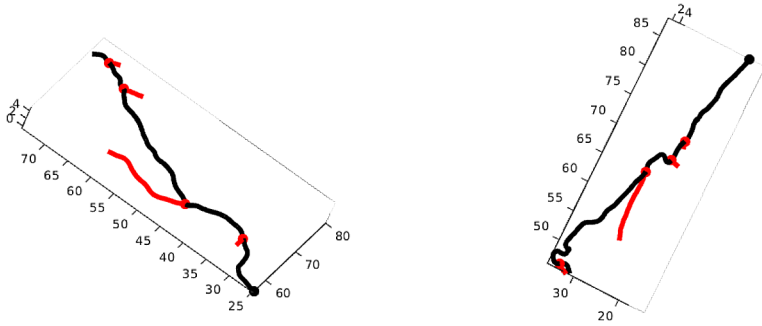
By design, this action does not affect the s_k . Arc-lengths are preserved under rotation and reparameterization, so branches of \boldsymbol{q} and $O(\boldsymbol{q}, \boldsymbol{\gamma})$ have the same spatial locations relative to the overall shape of the tree.

We treat this as a shape-preserving nuisance action, as we did for the group actions discussed in the shape space of curves (Section 2.1). We define a quotient space, \mathcal{Q}_n , on the closures of orbits under this action, which is roughly $\mathcal{P}_n / (\text{SO}(3) \times \Gamma^{n+1})$. We write $[\boldsymbol{q}] \in \mathcal{Q}_n$ for the closure of the orbit of \boldsymbol{q} . Distances in \mathcal{Q}_n are inherited from distances in \mathcal{P}_n . Given $\boldsymbol{q}^1, \boldsymbol{q}^2 \in \mathcal{P}_n$, the distance between their orbits is given by

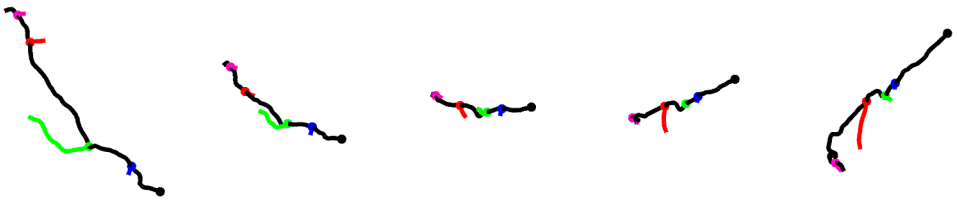
$$(12) \quad d'_n([\boldsymbol{q}^1], [\boldsymbol{q}^2]) = \inf\{d_n(\tilde{\boldsymbol{q}}^1, \tilde{\boldsymbol{q}}^2) : \tilde{\boldsymbol{q}}^1 \in [\boldsymbol{q}^1], \tilde{\boldsymbol{q}}^2 \in [\boldsymbol{q}^2]\}.$$

The following theorem (proved in the Appendix) gives sufficient conditions to minimize the preshape distance on the right-hand side of (12).

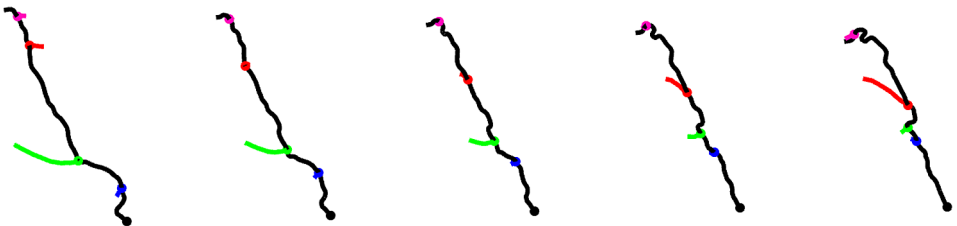
THEOREM 2. *Let $\boldsymbol{q}^1, \boldsymbol{q}^2 \in \mathcal{P}_n$ with branches composed of piecewise linear curves. Then there are $\tilde{\boldsymbol{q}}^1 \in [\boldsymbol{q}^1]$ and $\tilde{\boldsymbol{q}}^2 \in [\boldsymbol{q}^2]$ such that the prespace distance achieves the quotient distance: $d'_n([\boldsymbol{q}^1], [\boldsymbol{q}^2]) = d_n(\tilde{\boldsymbol{q}}^1, \tilde{\boldsymbol{q}}^2)$.*



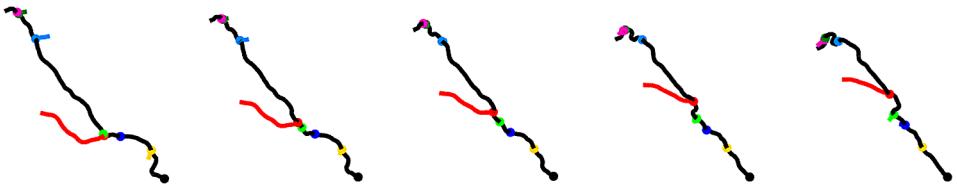
(a) Two trees $q^1, q^2 \in \mathcal{P}_4$. Below are three geodesic paths between these trees—each sampled at five equidistant points along the path and each in a different space. Distances and geodesics are computed using parameter values $\lambda_m = 0.01$, $\lambda_s = 0.01$, and $\lambda_p = 1.0$ [for use in equation (9)].



(b) Geodesic path between q^1, q^2 in \mathcal{P}_4 , the initial preshape space. The square distance given in equation (9) is $d_4(q^1, q^2)^2 = 1.8641$.



(c) Geodesic path between optimally-aligned representatives of $[q^1], [q^2] \in \mathcal{Q}_4$. The square distance given in equation (12) is $d'_4([q^1], [q^2])^2 = 0.7723$.



(d) Geodesic path between $\llbracket q^1 \rrbracket, \llbracket q^2 \rrbracket \in \tilde{\mathcal{Q}}$. The square distance is the lowest yet, with $d(\llbracket q^1 \rrbracket, \llbracket q^2 \rrbracket)^2 = 0.3619$.

FIG. 4. Examples of geodesic paths between the same two trees in different preshape and shape spaces.

Under the same conditions, this implies positivity of the quotient distance.

COROLLARY 1. *Let $\mathbf{q}^1, \mathbf{q}^2$ be as in Theorem 2. If $[\mathbf{q}^1] \neq [\mathbf{q}^2]$, then $d'_n([\mathbf{q}^1], [\mathbf{q}^2]) \neq 0$.*

PROOF. Suppose $d'_n([\mathbf{q}^1], [\mathbf{q}^2]) = 0$. Then there are $\tilde{\mathbf{q}}^1 \in [\mathbf{q}^1]$ and $\tilde{\mathbf{q}}^2 \in [\mathbf{q}^2]$ such that $d_n(\tilde{\mathbf{q}}^1, \tilde{\mathbf{q}}^2) = 0$, which implies $\tilde{\mathbf{q}}^1 = \tilde{\mathbf{q}}^2$, which implies $[\mathbf{q}^1] = [\mathbf{q}^2]$. \square

If we have such minimizing $\tilde{\mathbf{q}}^1, \tilde{\mathbf{q}}^2$, then we can form a geodesic path in \mathcal{P}_n by linearly interpolation as we did in (10). This path's length realizes the quotient distance, d'_n , and by taking the closed orbit of each point on this path, we form a corresponding path from $[\mathbf{q}^1]$ to $[\mathbf{q}^2]$ in \mathcal{Q}_n :

$$(13) \quad [\alpha(r)] = [(1 - r)\tilde{\mathbf{q}}^1 + r\tilde{\mathbf{q}}^2].$$

We refer to such minimal paths as geodesic paths in \mathcal{Q}_n . Figure 4(c) depicts a geodesic path between trees in \mathcal{Q}_4 which represent the equivalence classes of the same trees shown in Figure 4(b). The path from the first tree to the second appears to require much less deformation and the intermediate trees better resemble the endpoints. Indeed, the preshape distance between the aligned orbit representatives is much less than the preshape distance between the original unaligned trees: $d'_4([\mathbf{q}^1], [\mathbf{q}^2])^2 = 0.7723$ whereas $d_4(\mathbf{q}^1, \mathbf{q}^2)^2 = 1.8641$.

There is still one more shape-preserving equivalence that we wish to remove. Notice that in Figure 4(c), the endpoints $\tilde{\mathbf{q}}^1$ and $\tilde{\mathbf{q}}^2$ each have one prominent side branch and three very small ones. Specifically, \tilde{q}_2^1 and \tilde{q}_3^2 are much longer than the rest of the side branches in the two trees. When the trees are rotationally aligned, these branches protrude from the main branch in similar directions. A more satisfying correspondence between the two trees might be one where these two side branches are matched to each other. The geodesic path would deform one into the other, rather than deforming each into a diminutive branch as in Figure 4(c).

But under the framework so far, we only compare side branches which have the same given index. This was a choice of convenience which allowed us to focus on constructing the preshape spaces from the geometries of the trees' branches. The next section introduces an additional equivalence relation, which will give us our final shape space of trees.

2.4. *Trees with different numbers of side branches.* Since we wish to consider trees with an arbitrary number of side branches, we need a way to compare trees with different numbers of side branches. Such trees lie in different \mathcal{P}_n , and their shape classes are in different \mathcal{Q}_n . We define combined spaces by the following disjoint unions:

$$\mathcal{P} \equiv \bigsqcup_{n=0}^{\infty} \mathcal{P}_n, \quad \mathcal{Q} \equiv \bigsqcup_{n=0}^{\infty} \mathcal{Q}_n.$$

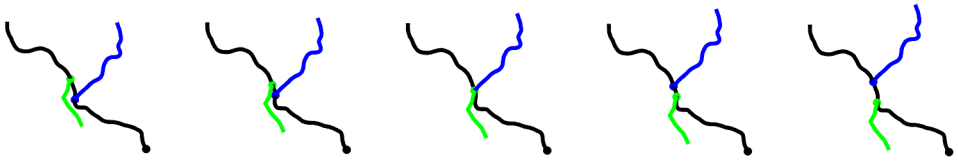


FIG. 5. Geodesic path between two very similar trees. The similarity is best understood if we can swap the order of branches when comparing them.

Each $\mathcal{P}_n, \mathcal{Q}_n$ has distances and geodesics defined above in equations (9) to (13). \mathcal{P} and \mathcal{Q} , however, are explicitly not connected and we cannot define distances between trees with different numbers of side branches since they lie in different components of \mathcal{P} and \mathcal{Q} . To get around this, we define an equivalence relation which glues these components together. In other words, the equivalence classes contain trees with different numbers of branches. Before explicitly stating the definition, we motivate it by describing two ways trees can be distinct in \mathcal{Q} , but occupy the same set of points in \mathbb{R}^3 .

One way for trees to have the same shape but different representations in \mathcal{Q} is the presence of zero-length branches. Our framework allows for side branches which are constant and do not emanate from the main branch. That is, a tree may have a side branch such that $\forall t \in [0, 1], \beta_k(t) = \beta_0(t_k)$ which is equivalent to $\forall t, q_k(t) = 0$. Practically speaking, one may intuitively view such a “branch” as a location where no branch exists and we call these *null* branches.

Another way for same-shaped trees to differ in \mathcal{Q} is differences in branch index order. That is, two trees may have collections of branches which are identical in shape and location, but are indexed in a different order. Intuitively, a pair of trees like this have the same shape, but our framework so far represents them differently and they have nonzero distance equations (9) and (12). An obvious way we could avoid this representational redundancy would be to add a constraint requiring side branches to be indexed in a pre-specified order (e.g., sorted by branch location parameter). We choose not to add such a constraint because it would preclude the existence of paths in tree-space where side branch base locations pass through each other and change order. Figure 5 illustrates a case where it may be preferable to include such paths in the space of trees.

Null branches and side branch order redundancy are dealt with by the following equivalence relation on \mathcal{P} .

DEFINITION 1. Given two SRVF trees, $\mathbf{q}^1 \in \mathcal{P}_{n_1}$ and $\mathbf{q}^2 \in \mathcal{P}_{n_2}$, we say they are *branch-equivalent* (denoted $\mathbf{q}^1 \sim \mathbf{q}^2$) if there are $\tilde{\mathbf{q}}^1 \in [\mathbf{q}^1], \tilde{\mathbf{q}}^2 \in [\mathbf{q}^2]$ such that:

1. Their main branches are the same: $\tilde{q}_0^1 = \tilde{q}_0^2$.
2. Nonnull side branches have the same shape and location (but not necessarily the same order): There exists $\sigma \in \mathcal{S}_n$, a finite permutation of order $n \geq \max\{n_1, n_2\}$, such that

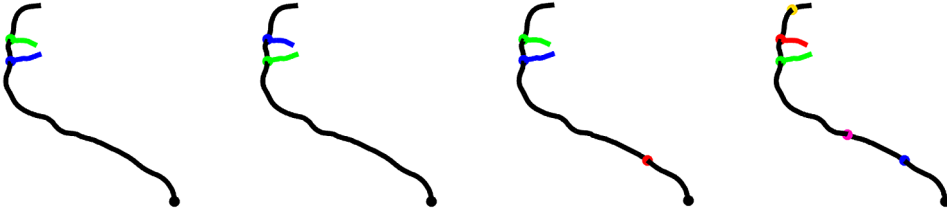


FIG. 6. Example of a set of branch-equivalent trees. This figure is best viewed in color. Branches with the same index in their respective tree are drawn in the same color. The two leftmost trees have two branches each, which are identical, but with swapped index order. The two rightmost have the same nonnull branches, and have some null branches inserted at various places.

- (a) $(\tilde{q}_k^1, \tilde{s}_k^1) = (\tilde{q}_{\sigma^{-1}(k)}^2, \tilde{s}_{\sigma^{-1}(k)}^2)$ for any k where $\tilde{q}_k^1 \neq 0$;
- (b) $(\tilde{q}_k^2, \tilde{s}_k^2) = (\tilde{q}_{\sigma(k)}^1, \tilde{s}_{\sigma(k)}^1)$ for any k where $\tilde{q}_k^2 \neq 0$.

Given $\mathbf{q} \in \mathcal{P}_n$ (for some n), we denote its equivalence class under \sim by $[[\mathbf{q}]]$, and we denote the quotient space $\tilde{\mathcal{Q}} = \mathcal{P}/\sim$.

It is obvious that $[\mathbf{q}] \subset [[\mathbf{q}]]$ for any $\mathbf{q} \in \mathcal{P}$. In other words, $\tilde{\mathcal{Q}}$ is a strictly coarser partition than \mathcal{Q} .

This definition allows us to insert and remove an arbitrary number of null branches into a SRVF tree while maintaining branch-equivalence. Hence an equivalence class in $\tilde{\mathcal{Q}}$ has representatives in \mathcal{P}_n for infinitely many n . More specifically, if $\mathbf{q} \in \mathcal{P}$ and \mathbf{q} has exactly K nonnull side branches, then $[[\mathbf{q}]]$ intersects \mathcal{P}_n for all $n \geq K$. In such a case, we say K is the *order* of $[[\mathbf{q}]]$. If K is the order of $[[\mathbf{q}]]$ and $\tilde{\mathbf{q}} \in \mathcal{P}_K \cap [[\mathbf{q}]]$ —in other words $\tilde{\mathbf{q}}$ has no null side branches—then we say $\tilde{\mathbf{q}}$ is *minimal*. We carry these definitions back to the original trees: β^1 and β^2 are branch-equivalent if and only if their SRVF representations $\mathbf{q}^1 \sim \mathbf{q}^2$ are branch-equivalent; the order of tree β is the order of its SRVF representation; β is minimal if and only if its SRVF representation is minimal.

Figure 6 shows a set of branch-equivalent trees of order 2. From left to right, call them β^1 , β^2 , β^3 , and β^4 . The original tree, β^1 , is a real axonal tree, taken directly from the IBV dataset (described in Section 3) and the others are generated from β^1 by permuting the indices of side branches and/or inserting null branches. Like β^1 , β^2 is minimal and is formed by swapping the indices of the original: $\beta_1^2 = \beta_2^1$ and $\beta_2^2 = \beta_1^1$. Next, β^3 has the same β_1, β_2 as β^1 and has an additional null-branch, β_3^3 , which appears as a red dot. The rightmost tree, β^4 , has three additional null branches compared to the original tree, and the indices are permuted so that $\beta_2^4 = \beta_1^1$ and $\beta_3^4 = \beta_1^1$ while β_1^4, β_4^4 , and β_5^4 are the null branches. These four trees all belong to the same branch-equivalence class because the set of nonnull side-branches is the same.

The quotient distance under \sim is

$$(14) \quad d([[q^1]], [[q^2]]) = \inf\{d'_n([\tilde{q}^1], [\tilde{q}^2]) : [\tilde{q}^1] \subset [[q^1]], [\tilde{q}^2] \subset [[q^2]]\}.$$

In the right-hand side of equation (14), a candidate pair of representatives $\tilde{q}^1 \in \llbracket q^1 \rrbracket$ and $\tilde{q}^2 \in \llbracket q^2 \rrbracket$ is called a *matching*. If \tilde{q}^1, \tilde{q}^2 minimize the distance, $d'_n(\llbracket \tilde{q}^1 \rrbracket, \llbracket \tilde{q}^2 \rrbracket)$, then we call it an *optimal matching* between $\llbracket q^1 \rrbracket$ and $\llbracket q^2 \rrbracket$ (or between q^1 and q^2). Implicit in the definition of this distance is that the representatives which form a matching must be in the same \mathcal{P}_n . If \tilde{q}^1, \tilde{q}^2 are an optimal matching, then the (straight line) geodesic between them in \mathcal{P} has length equal to the quotient distance. Taking the equivalence class of each point on this path, we get a path from $\llbracket q^1 \rrbracket$ to $\llbracket q^2 \rrbracket$, that we call a geodesic path in $\tilde{\mathcal{Q}}$:

$$(15) \quad \llbracket \alpha(r) \rrbracket = \llbracket (1-r)\tilde{q}^1 + r\tilde{q}^2 \rrbracket.$$

Figure 4(d) is the satisfying result for the example studied in Figure 4. The large side branches are deformed into one another and the tiny side branches are shrunk to null. Compared to the previous two geodesics, the path in Figure 4(d) looks more natural and leads to a sharp reduction in square distance between the two tree shapes.

Building on Theorem 2, the existence of an optimal matching can be guaranteed under some mild assumptions about the curves that make the branches. The following theorem is proved in the [Appendix](#).

THEOREM 3. *Let $q^1 \in \mathcal{P}_{n_1}$ and $q^2 \in \mathcal{P}_{n_2}$ with branches composed of piecewise linear curves. Then there are $\tilde{q}^1 \in \llbracket q^1 \rrbracket$ and $\tilde{q}^2 \in \llbracket q^2 \rrbracket$ such that the pre-space distance achieves the quotient distance: $d(\llbracket q^1 \rrbracket, \llbracket q^2 \rrbracket) = d_n(\tilde{q}^1, \tilde{q}^2)$. Furthermore, such \tilde{q}^1 and \tilde{q}^2 exist in \mathcal{P}_n for some $n \leq n_1 + n_2$.*

The first part (existence) implies positivity of the distance between distinct shape classes.

COROLLARY 2. *Let q^1, q^2 be as in Theorem 3. If $\llbracket q^1 \rrbracket \neq \llbracket q^2 \rrbracket$, then $d(\llbracket q^1 \rrbracket, \llbracket q^2 \rrbracket) \neq 0$.*

PROOF. Suppose $d(\llbracket q^1 \rrbracket, \llbracket q^2 \rrbracket) = 0$. Then there are $\tilde{q}^1 \in \llbracket q^1 \rrbracket$ and $\tilde{q}^2 \in \llbracket q^2 \rrbracket$ such that $d_n(\tilde{q}^1, \tilde{q}^2) = 0$, which implies $\tilde{q}^1 = \tilde{q}^2$, which implies $\llbracket q^1 \rrbracket = \llbracket q^2 \rrbracket$. \square

Computing such an optimal matching (or its approximation) is the subject of the next section.

2.5. Computing an optimal matching between two trees. The second part of Theorem 3 helps us narrow our search for an optimal matching. Given $q^1 \in \mathcal{P}_{n_1}$ and $q^2 \in \mathcal{P}_{n_2}$, we need only consider branch-equivalent trees with $N = n_1 + n_2$

branches. We can formulate the distance between $[[\mathbf{q}^1]]$ and $[[\mathbf{q}^2]]$ [equation (14)] in a more computationally-motivated way:

$$(16) \quad \begin{aligned} & d([[q^1]], [[q^2]])^2 \\ &= \inf_{\substack{O \in \text{SO}(3) \\ \boldsymbol{\gamma} \in \Gamma^{N+1} \\ \sigma \in \mathcal{S}_N}} \{d_N(\tilde{\mathbf{q}}^1, O(\tilde{\mathbf{q}}_\sigma^2, \boldsymbol{\gamma}))\}^2 : \tilde{\mathbf{q}}^1 \in \mathcal{Q}_N^1, \tilde{\mathbf{q}}^2 \in \mathcal{Q}_N^2, \end{aligned}$$

where

$$\begin{aligned} \mathcal{Q}_N^i &= \{\tilde{\mathbf{q}}^i \in [[\mathbf{q}^i]] \cap \mathcal{P}_N : (\tilde{q}_k^i, \tilde{s}_k^i) = (q_k^i, s_k^i) \text{ for } k = 1, \dots, n_i\}, \\ \mathbf{q}_\sigma &= (q_0, \{q_{\sigma(k)}, s_{\sigma(k)}\}). \end{aligned}$$

In words, \mathcal{Q}_N^i is the set of trees we can construct by starting with $\mathbf{q}^i \in \mathcal{P}_{n_i}$, and inserting $N - n_i$ null branches, \tilde{q}_k^i , at arbitrary locations, s_k , for $k = n_i + 1, \dots, N$. The subscript σ denotes a permutation of the side branch indices. To construct the matching using this setup, we add null branches to the two trees to get $\tilde{\mathbf{q}}^1, \tilde{\mathbf{q}}^2$, then transform $\tilde{\mathbf{q}}^2$ by applying some $O \in \text{SO}(3)$, $\boldsymbol{\gamma} \in \Gamma^{n+1}$, and $\sigma \in \mathcal{S}_N$. When we add null branches to form $\tilde{\mathbf{q}}^1, \tilde{\mathbf{q}}^2$, the new s_k are the only additional information, and it turns out the best s_k are determined by the choice of permutation. The challenge here is to optimize over all these transformation spaces simultaneously. Ideally, we would like to find $O, \boldsymbol{\gamma}, \sigma$ that achieve the infimum, and thus give us an optimal matching. It may not be possible to find the optimal matching this way for reasons mentioned in Section 2.1. However, we can use this approach to find a matching so that the distance, d_N , in the right-hand side of equation (16) is arbitrarily close to the infimum.

To optimize over this daunting list of spaces— $\text{SO}(3)$, Γ^{n+1} , and \mathcal{S}_N —we use an alternating optimization approach. We repeatedly optimize with respect to:

- (1) $O \in \text{SO}(3)$ (with fixed parameterizations and permutation), then
- (2) $\boldsymbol{\gamma} \in \Gamma^{N+1}$ and $\sigma \in \mathcal{S}_N$ (with fixed rotation).

Finding an optimal $O \in \text{SO}(3)$ is easy—it is the Procrustes rotation problem. The novel part is step (2): the simultaneous optimization over reparameterization and permutation. Most of the rest of this section describes how we accomplish this second task, and the process is given more explicitly in Algorithm 2. The top-level alternating optimization procedure is given in Algorithm 1.

One part of the reparameterization-rotation can be separated from the rest, namely γ_0 . In the expression for the pre-space distance [equation (9)], reparameterization of the main branch will only affect the first term—the main branch shape term. Likewise, reparameterization of side branches only affects the second term, which measures side branch shape distance. Permutation of side branches affects both the last two terms, which measure shape difference due to side branch shape and side branch position. Thus, we can optimize for $\gamma_0 \in \Gamma$ separately from $\sigma \in \mathcal{S}_N$

Algorithm 1 Top-level optimization over branch-equivalence**Input:**

- $q^1 \in \mathcal{P}_{n_1}, q^2 \in \mathcal{P}_{n_2}$ —trees to be aligned
 $\lambda = (\lambda_m, \lambda_s, \lambda_p)$ —prespace distance weight parameter vector
 I_{\max} —Number of iterations.

Output:

- $\tilde{q}^1 \in \mathcal{Q}_n^1, \tilde{q}^2 \in \mathcal{Q}_n^2$ —same as q^1, q^2 but with additional null branches
 $O \in \text{SO}(3)$ —optimal rotation matrix
 $\gamma \in \Gamma^{n+1}$ —optimal set of curve reparameterizations
 $\sigma \in \mathcal{S}_n$ —optimal permutation of n side branches

- 1: **procedure** ALIGNFULL
- 2: $\tilde{q}^1, \tilde{q}^2, \gamma, \sigma \leftarrow \text{REPARAMPERMUTE}(O_{\text{id}}, q^1, q^2, \lambda)$
- 3: **for** I_{\max} iterations **do**
- 4: $O \leftarrow \text{PROCRUSTES}(\tilde{q}^1, (\tilde{q}^2, \gamma), \lambda)$
- 5: $\tilde{q}^1, \tilde{q}^2, \gamma, \sigma \leftarrow \text{REPARAMPERMUTE}(O, \tilde{q}^1, \tilde{q}^2, \lambda)$
- 6: **end for**
- 7: **end procedure**

and $\{\gamma_1, \dots, \gamma_N\} \in \Gamma^N$. But the latter two are still entangled with each other—the choice of side reparameterizations depends on which branches are matched to each other; the choice of permutation depends on knowing the shape distance between side branches, which requires them to be optimally aligned.

To compute this combined optimization, we frame it in terms of the linear assignment problem. We build an $N \times N$ matrix, \mathbf{E} , of matching costs between the side branches of q^1 and q^2 . That is, if σ is chosen so that $\sigma(i) = j$, then the i th branch of q^1 is matched to the j th branch of q^2 . This matched pair will make some contribution to the squared preshape distance in the right-hand side of equation (16) and this contribution is stored in entry E_{ij} of the cost matrix. Furthermore, these contributions are additive over the set of matched pairs and the matrix is constructed in a way so that a given choice of σ determines the optimal side reparameterizations, $\gamma_1, \dots, \gamma_N$. Thus, the side branch shape and location terms of the preshape distance in equation (16) are completely determined by $\sum_i E_{i, \sigma(i)}$. Once this matrix \mathbf{E} is formed, we can find the optimal permutation by minimizing this sum. This is known as the linear assignment problem—a well-studied computational task which we can solve with fast off-the-shelf methods with worst-case time complexity $O(N^3)$. We use an implementation of the method proposed in Jonker and Volgenant (1987), which we refer to as LAPJV.

The dimensions of \mathbf{E} exceed the number of branches in q^1 and q^2 , so entry E_{ij} corresponds to a matching between nonnull branches if and only if $i \leq n_1$ and $j \leq n_2$. Higher indices indicate matchings that involve the null branches added to form \tilde{q}^1 and \tilde{q}^2 . With this distinction in mind, we divide the cost matrix into

Algorithm 2 Simultaneous optimization d_N with respect to $\boldsymbol{\gamma} \in \Gamma^{N+1}$, $\sigma \in \mathbf{S}_N$

Input:

$O \in \text{SO}(3)$ —iven rotation matrix
 $\mathbf{q}^1 \in \mathcal{P}_{n_1}$, $\mathbf{q}^2 \in \mathcal{P}_{n_2}$ —trees to be aligned
 $\boldsymbol{\lambda} = (\lambda_m, \lambda_s, \lambda_p)$ —pre-space distance weight parameter vector

Output:

$\tilde{\mathbf{q}}^1 \in \mathcal{Q}_n^1$, $\tilde{\mathbf{q}}^2 \in \mathcal{Q}_n^2$ —same as \mathbf{q}^1 , \mathbf{q}^2 but with additional null branches
 $\boldsymbol{\gamma} \in \Gamma^{n+1}$ —optimal set of curve reparameterizations
 $\sigma \in \mathbf{S}_n$ —optimal permutation of N side branches
 E_{match} —energy (square distance) for optimal $\boldsymbol{\gamma}$, σ

```

1: procedure REPARAMPERMUTE
2:    $N \leftarrow n_1 + n_2$ 
3:    $\mathbf{q}^2 \leftarrow \hat{O}\mathbf{q}^2$ 
4:   for  $i$  from 1 to  $n_1$  do
5:     for  $j$  from 1 to  $n_2$  do
6:        $\tilde{\gamma}_{ij} \leftarrow \text{DYNAMICPROGRAMMINGQ}(q_i^1, q_j^2)$ 
7:        $E_{ij} \leftarrow \lambda_m \|q_i^1 - (q_j^2, \tilde{\gamma}_{ij})\|^2 + \lambda_p (s_i^1 - s_j^2)^2$ 
8:     end for
9:      $E_{ij} \leftarrow \lambda_s \|q_i^1\|^2$  for  $n_2 < j \leq N$ 
10:  end for
11:  for  $j$  from 1 to  $n_2$  do
12:     $E_{ij} = \lambda_s \|q_j^2\|^2$  for  $n_1 < i \leq N$ 
13:  end for
14:   $E_{ij} \leftarrow 0$  for  $n_1 < i \leq N$  and  $n_2 < j \leq N$ 
15:   $\sigma, E_{\text{sides}} \leftarrow \text{LAPJV}(E)$ 
16:   $I \leftarrow \{i \mid 1 \leq i \leq n_1 \text{ or } 1 \leq \sigma(i) \leq n_2\}$ 
17:   $\sigma \leftarrow \sigma|_I$ 
18:   $n \leftarrow |I|$ 
19:   $\gamma_0 \leftarrow \text{DYNAMICPROGRAMMINGQ}(q_0^1, q_0^2)$ 
20:  for  $i$  from 1 to  $n_1$  do
21:    if  $1 \leq \sigma(i) \leq n_2$  then
22:       $\gamma_i = \tilde{\gamma}_{i, \sigma(i)}$ 
23:    else
24:       $\gamma_i = \gamma_{id}$ 
25:    end if
26:  end for
27:   $\gamma_i \leftarrow \gamma_{id}$  for  $n_1 < i \leq n$ 
28:   $E_{\text{match}} \leftarrow \lambda_m \|q_0^1 - (q_0^2, \gamma_0)\|^2 + E_{\text{sides}}$ 
29:   $\mathbf{q}^2 \leftarrow O^{-1}\mathbf{q}^2$ 
30:   $\tilde{\mathbf{q}}^1 \leftarrow \mathbf{q}^1$ 
31:   $(\tilde{q}_i^1, \tilde{s}_i^1) \leftarrow (0, s_{\sigma(i)}^2)$  for  $n_1 < i \leq n$ 
32:   $\tilde{\mathbf{q}}^2 \leftarrow \mathbf{q}^2$ 
33:   $(\tilde{q}_j^2, \tilde{s}_j^2) \leftarrow (0, s_{\sigma^{-1}(j)}^1)$  for  $n_2 < j \leq n$ 
34: end procedure
    
```

\triangleright Discard null-to-null matches
 $\triangleright \sigma$ is now in \mathbf{S}_n for some n
 \triangleright s.t. $\max\{n_1, n_2\} \leq n \leq n_1 + n_2$

named submatrices as follows:

$$(17) \quad \mathbf{E} = \begin{bmatrix} \mathbf{M} & \mathbf{U}^1 \\ \mathbf{U}^2 & \mathbf{0} \end{bmatrix}.$$

In the upper left is the $n_1 \times n_2$ matrix of costs to match nonnull branches to each other, which we call \mathbf{M} . Upper right is the $n_1 \times n_1$ matrix, \mathbf{U}^1 , of costs to kill the side branches of \mathbf{q}^1 (i.e., match them to newly-added null branches of $\tilde{\mathbf{q}}^2$). Similarly, lower left is the $n_2 \times n_2$ matrix, \mathbf{U}^2 , of costs to kill the side branches of \mathbf{q}^2 . Lower right is an $n_2 \times n_1$ matrix of zeros. Since the entries of \mathbf{M} and the \mathbf{U}^j correspond to qualitatively different kinds of matching, we compute the costs in different ways.

First, we describe the submatrices involving null branches. \mathbf{U}^1 sits in the first n_1 rows and last n_1 columns of \mathbf{E} . Element U^1_{ij} is the cost of creating a null branch, $\tilde{q}^2_{n_2+j}$, and matching it to q^1_i . Since we can insert the null branch at an arbitrary location, we can choose its location to have the same relative arc-length on the main branch: $s^2_{n_2+j} = s^1_i$. That way, this matched pair contributes nothing to the position term of d_N (any other choice would have a positive cost), so the only possible contribution remaining is to the side branch shape term. The null branch SRVF is identically zero, so the contribution is $U^1_{ij} = \lambda_s \|\tilde{q}^1_i\|^2$. Similarly, if we create a null $\tilde{q}^1_{n_1+i}$ and match it to nonnull q^2_j , the cost is $U^2_{ij} = \lambda_s \|q^2_j\|^2$. In both cases, the square norm the SRVF is invariant to reparameterization. Thus, if LAPJV chooses any $E_{i,\sigma(i)}$ from \mathbf{U}^1 or \mathbf{U}^2 , then any $\gamma_i \in \Gamma$ gives the same cost. For simplicity, we use $\gamma_i = \gamma_{id}$ in this case.

Note that the entries of \mathbf{U}^1 do not depend on j , which means each of its n_1 rows is the same value repeated n_1 times, and its columns are identical to each other. Similarly, the columns of \mathbf{U}^2 consist of repeated values and its rows are identical. This may seem redundant, but it is necessary for the linear assignment problem to be set up correctly. Each row can only be matched to one column and vice versa because the permutation, σ , is bijective. For a given pair of trees, it may be that the optimal matching has all n_1 side branches of \mathbf{q}^1 matched to null branches and the same for all n_2 side branches of \mathbf{q}^2 . In this most extreme case, LAPJV will choose σ so that every $E_{i,\sigma(i)}$ is in \mathbf{U}^1 and \mathbf{U}^2 . To accommodate this, both of these submatrices must be square, which is precisely how we construct \mathbf{E} .

These extra entries create an empty space in the lower right $n_2 \times n_1$ submatrix of \mathbf{E} , where the entries correspond to no branches on either tree. For any such matching, the branches are null, and we can choose $s^1_i = s^2_j$, so the matching contribute no cost to d_N , and thus, the whole submatrix is filled with zeros. In the case that those entries are chosen by the linear assignment solver, we just discard them afterward, and ultimately get a matching in \mathcal{P}_n for some n such that $\max\{n_1, n_2\} \leq n \leq N$.

Finally, we come to the upper left $n_1 \times n_2$ submatrix \mathbf{M} , in which element M_{ij} is the cost of matching q^1_i to q^2_j [i.e., the cost of setting $\sigma(i) = j$]. This choice of

matching contributes to the last two terms of d_N , so its cost is a weighted sum of the shape distance between the SRVFs, and the sliding cost due to difference in relative location:

$$(18) \quad \mathbf{M}_{ij} = \lambda_m \|q_i^1 - (q_j^2, \tilde{\gamma}_{ij})\|^2 + \lambda_p (s_i^1 - s_j^2)^2.$$

Here, $\tilde{\gamma}_{ij}$ is the optimal reparameterization between the two SRVFs: $\tilde{\gamma}_{ij} = \operatorname{argmin}_{\gamma \in \Gamma} \|q_i^1 - (q_j^2, \gamma)\|^2$. This means we must find an optimal reparameterization between two curves for each element of \mathbf{M} , which we compute using the same dynamic programming algorithm, as in [Srivastava et al. \(2011\)](#). So in addition to constructing \mathbf{M} , we are constructing a $n_1 \times n_2$ matrix of reparameterization functions, $\tilde{\gamma}_{ij}$. Then, for any matching that includes $\mathbf{M}_{i,\sigma(i)}$, the corresponding reparameterization is $\gamma_i = \tilde{\gamma}_{i,\sigma(i)}$.

The dynamic programming algorithm mentioned above has quadratic time complexity with respect to the number of points in the curve discretization. We invoke this algorithm $n_1 n_2$ times to construct \mathbf{M} . If each branch is discretized using T points, then constructing \mathbf{M} takes $O(n_1 n_2 T^2)$ time. In practice, this is the most computationally expensive part of [Algorithm 2](#), although in principle, LAPJV has worst case time complexity $O(N^3)$, which could take longer for large enough n_1, n_2 and small enough T .

2.6. Example alignments. [Figure 7](#) gives example geodesic paths between two trees which each have one side branch. The two geodesics illustrate the effect of varying the tuning parameters. For this simple case, there are two choices of side branch matching: they can be matched to each other or they can each be matched to a null branch. For relatively high values of the sliding parameter and low values of the side branch shape parameter, it is too expensive to slide them to meet each other and little cost is saved by matching them to each other instead of shrinking/growing them. The branches go unmatched for such parameters. Conversely, if side branch shape is influential and side branch position is not, then the branches get matched by the algorithm.

[Figure 8](#) shows a slightly more complex case. It shows a tree with one branch and a tree with two. The side branch of the one-branch tree has better shape similarity to the branch farthest from it on the other tree. As the ratio λ_s/λ_p decreases, the single branch gets matched to the farther branch, then the nearer branch, then to nothing.

[Figure 9](#) shows an example where several branches in one tree are matched to several branches in the other. Moderate parameter values were chosen so some branches get matched and some do not. [Figure 10](#) shows two more examples with more complex branching structures.

3. Distance-based classification. In order to demonstrate that the proposed shape metric captures differences in neuron shape that correspond to biological

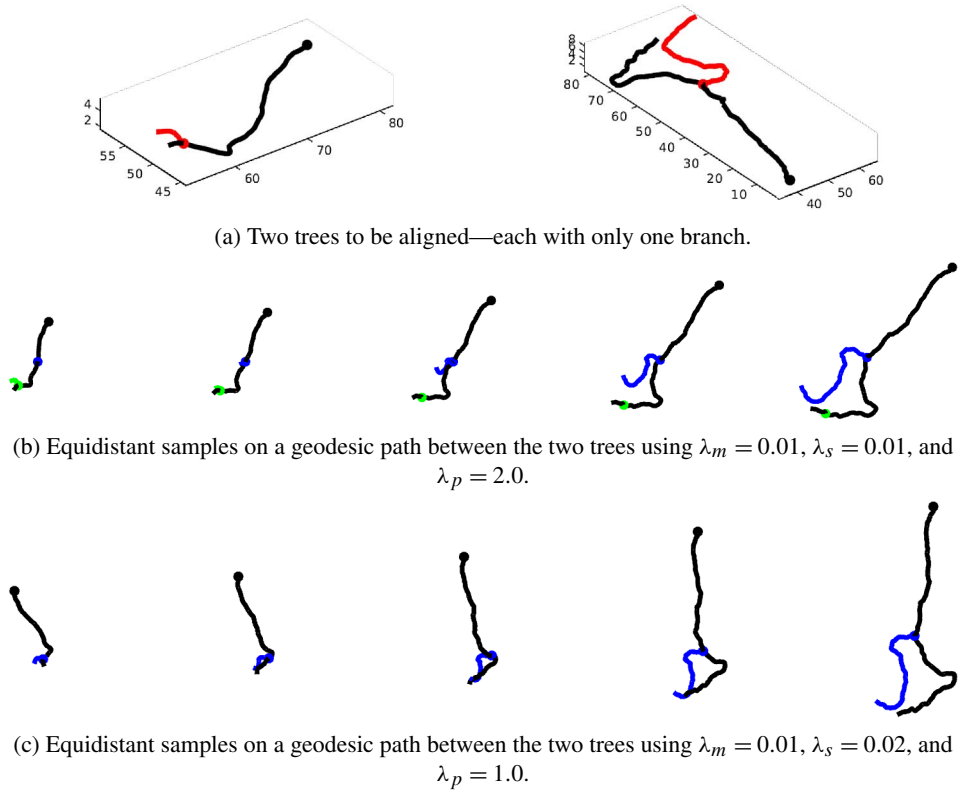


FIG. 7. Two simple trees and geodesics between them. In the first geodesic, the distance has higher weight on the side branch position (λ_p) and lower weight on the side branch shape (λ_s).

differences, we perform cross-validated classification on several real datasets. In each case, we compute the full pairwise distance matrix on the dataset and perform classifications based on those distances. We use the following three datasets.

IBV: 91 axonal trees extracted from confocal microscope images collected by biologists at Institute Of Biology Valrose (IBV). The axons belong to neurons in the mushroom bodies of *Drosophila Melanogaster* (fruit flies). The dataset is divided into three groups—wild type (control group) and two mutant types. In mutant type 1 flies, the gene which encodes *imp* is inactive and in mutant type 2, the gene which encodes *profilin* is inactive. In the dataset, there are 45 type 1 mutants, 15 type 2 mutants, and 31 wild type flies.

Wu: 41 apical dendrites taken from the CA1 region of the hippocampus in mice. We obtained these neuron reconstructions from NeuroMorpho.org [Suo et al. (2012)]. (We refer to it as the Wu dataset.) Similar to the IBV dataset, these neurons are divided into groups according to experimentally-controlled genetic type: 21 are taken from wild type mice, and 20 are from mice with a cluster of *protocadherin* genes knocked out.

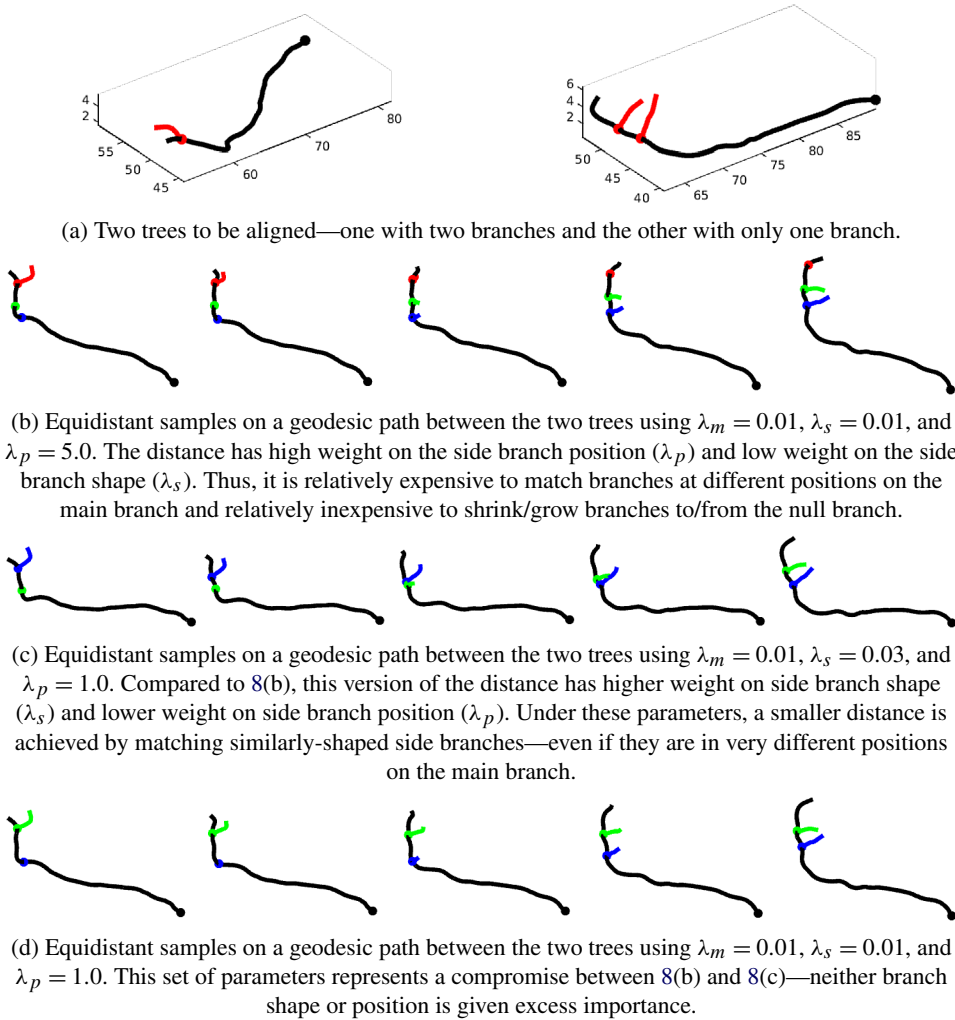


FIG. 8. Two simple trees and geodesics between them under three different sets of the parameter values, λ_m , λ_s , and λ_p .

Chen: 99 apical dendrites of pyramidal neurons taken from both the CA1 region of the hippocampus and layer V of the sensorimotor cortex in rats. These reconstructions are also from NeuroMorpho.org and are studied in [Chen et al. \(2014\)](#). This dataset has three experimental groups: (1) a group subjected to bile duct ligation (BDL), (2) a group subjected to bile duct ligation and fed a diet containing ammonia acetate (abbreviated BDLHD in the paper), and (3) a control group.

The IBV data was supplied to us in the format required by our method: a main branch with side branches protruding from it. We chose the two datasets

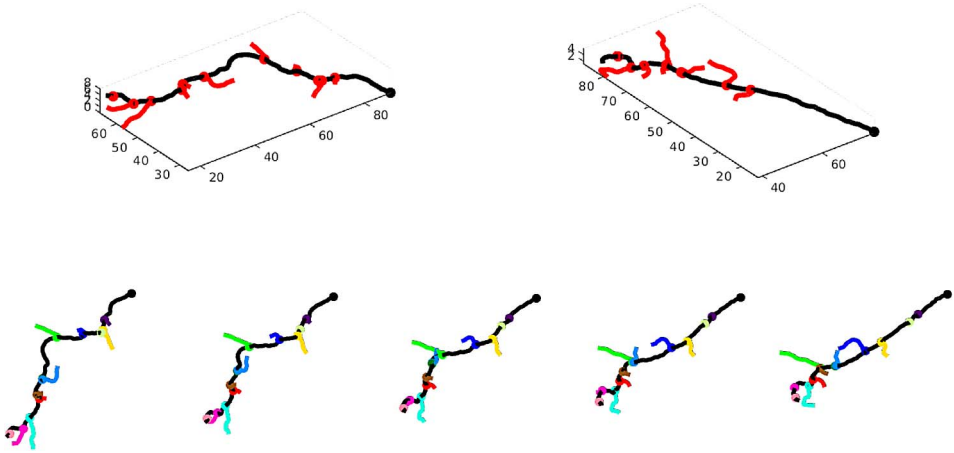


FIG. 9. Two trees from the IBV dataset with many side branches. Upper left is a tree with 11 side branches and upper right is a tree with 9. Below is a geodesic path between them using $\lambda_m = 0.01$, $\lambda_s = 0.01$, and $\lambda_p = 1.0$.

from Neuromorpho.org because the apical dendrites of those neurons appeared to mostly fit the topological requirements of our method. Neuromorpho.org does not label its data with our main-side distinction, so we have to have a way of choosing the main branch. To do so, we simply take the terminal point with the longest arc-length from the start of the tree and call that the main branch. For each bifurcation along the main branch, we take the longest path to a terminal point and call that a side branch. Any further branching is discarded. We chose the Wu and Chen data specifically because they contain trees for which little (and sometimes none) of

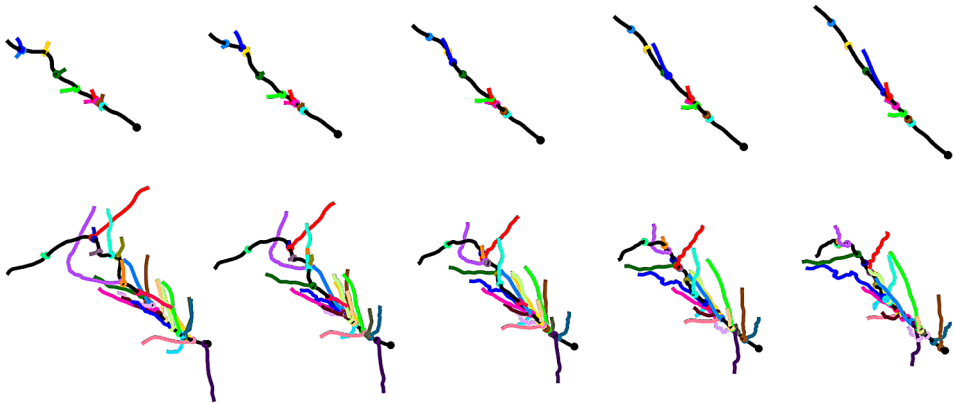


FIG. 10. Additional geodesic path examples. Top row is between two aligned trees from the Wu dataset using $\lambda = (0.02, 1.0, 1.0)$. Bottom row is between two from the Chen dataset using $\lambda = (0.01, 0.01, 1.0)$.

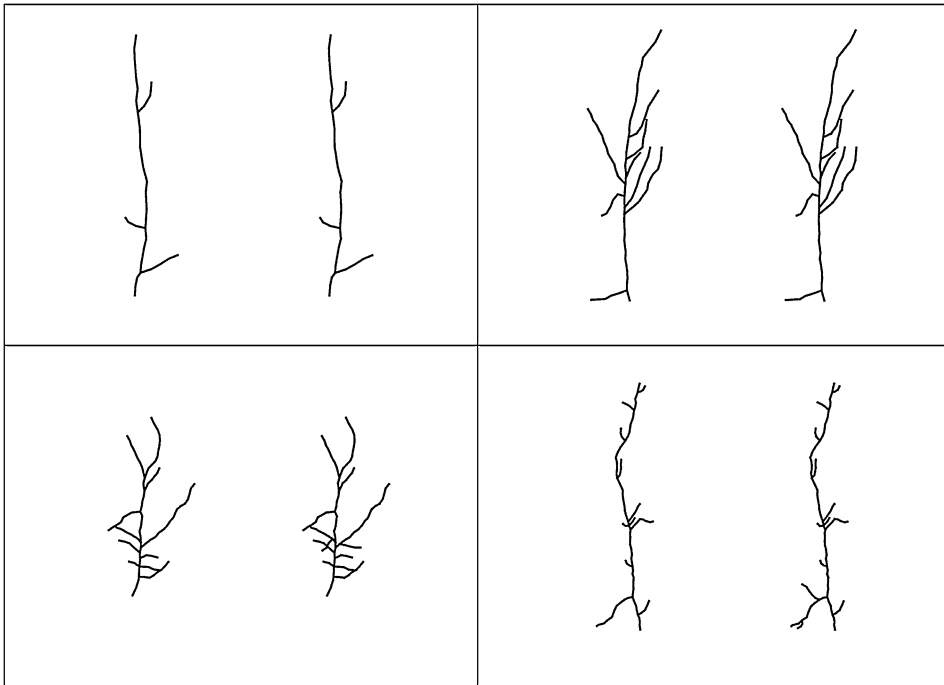


FIG. 11. Example topological simplifications of apical dendrites from the Wu dataset. In each pair, the neuron depicted on the left is the simplified structure required by our method. The one on the right shows the full branching structure contained in the *Neuromorpho.org* data. In the top two examples, the simplification causes no change and we do not have to discard any branches. In each of the bottom two examples, we only have to discard one or two small branches, and most of the branching structure is left intact.

the structure would be discarded this way. Figure 11 gives examples from the Wu dataset that illustrate the effect of this simplification.

There are many ways to perform classification based on a distance alone. Since the methodological focus of this paper is the construction of the distance itself, we simply use a common method: a SVM classifier with a Gaussian radial basis function (RBF) kernel. Using the distance defined in equation (14), the kernel function of two prespace trees is

$$(19) \quad K(\mathbf{q}_1, \mathbf{q}_2) = \exp(-\gamma \cdot d(\llbracket \mathbf{q}^1 \rrbracket, \llbracket \mathbf{q}^2 \rrbracket)^2).$$

A potential downside to this choice of classifier is that Gaussian kernels are not necessarily positive definite (PD) in non-Euclidean spaces. Feragen, Lauze and Hauberg (2015) shows that in geodesic metric spaces, Gaussian kernels are PD for all $\gamma > 0$ if and only if the space is Euclidean. Similarly, the classical result of Schoenberg (1938) shows that for general metric spaces, Gaussian kernels are PD for all $\gamma > 0$ if and only if the space can be isometrically mapped to an inner

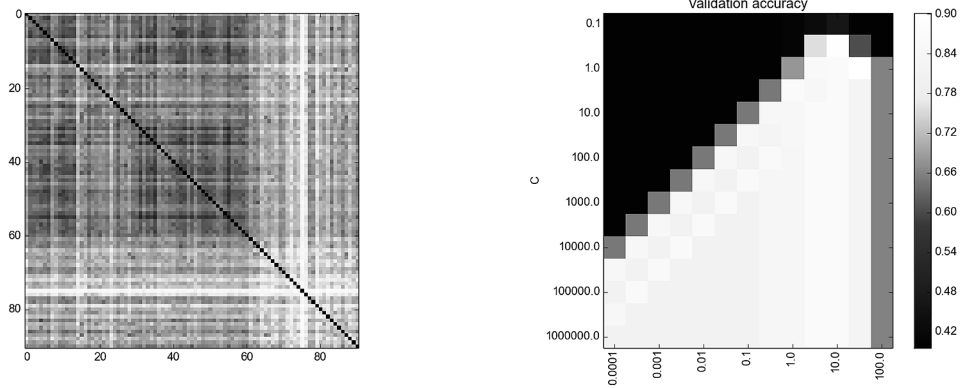


FIG. 12. *Left: Distance matrix of the 91 Drosophila axons in the IBV dataset, using $\lambda = (0.001, 0.001, 5.0)$. Indices 1 to 45 are mutant type 1, indices 46 to 60 are mutant type 2, and indices 61 to 91 are wild type. Right: Grid search of RBF/SVM parameters, γ and C , for CV success.*

product space. The former result does not directly apply because our metric is not a geodesic metric. It is not obvious whether we can apply the latter result—there is no obvious choice of inner product space that would preserve our quotient metric. We suspect the Gaussian kernel is not generally PD in our space, but as the examples in this section demonstrate, we are still able to get some good classification results using it. We leave the exploration of kernel methods in this space as a possible future direction of research.

3.1. IBV dataset classification. First, we perform stratified 5-fold cross-validated classification on the IBV dataset with the two mutation types combined as a single category, so the group sizes are $n = 60$ for the mutation group and $n = 31$ for the wild type group. Figure 12 shows the distance matrix for a particular choice of λ and the grid search of the kernel/SVM tuning parameters. For the best choice of parameters, this had a successful classification rate of 0.901 (82 of 91 correct).

3.2. Wu dataset classification. We do the same for the Wu dataset, but with a more exhaustive search over choices of λ . With $\lambda_p = 1.0$ fixed, we vary λ_m and λ_s over an approximately-evenly-spaced grid of values on a log scale, as shown in Figure 13(a). For each choice of λ , we compute the pairwise distance matrix then perform the grid search over RBF/SVM parameters to find the best stratified 5-fold CV success. The pairwise distance matrix for that best λ is shown in Figure 13. The highest-achieved success rate is 0.805 (33 of 41 correct), using $\lambda = (0.02, 1.0, 1.0)$.

3.3. Chen dataset classification. The Chen dataset is the largest one we look at and it also has the most categories. There are three experimental conditions and

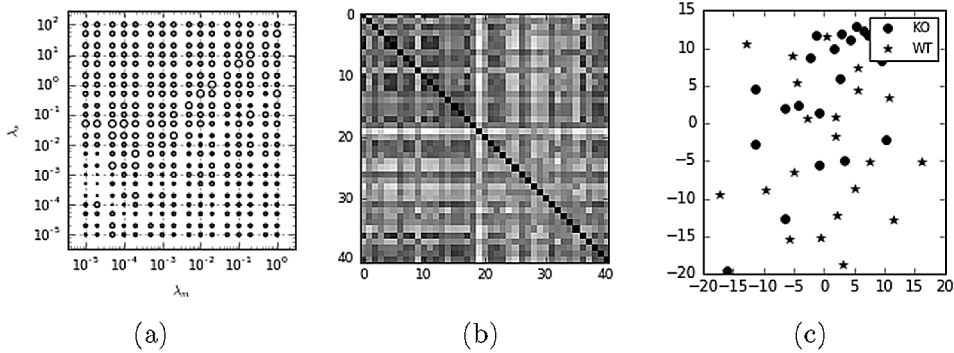


FIG. 13. Classification experiment on the 41 mouse dendrites in the Wu dataset. (a) Grid search over distance tuning weights. CV classification accuracy is depicted both by brightness and circle size. Best accuracy is 0.805, achieved with $\lambda = (0.02, 1.0, 1.0)$. (b) Pairwise distance matrix using the best λ . By index, the first 20 are in the knockout group and the remaining 21 are wild type. (c) MDS plot for the same distances.

two brain regions, and the dataset contains all six combinations with similarly-sized groups in each category (Table 1 gives the exact counts). We attempt to classify the neurons according to region only (2-class), experimental condition only (3-class), and the combination of both factors (6-class). In each case, we perform a procedure similar to what we did with Wu: we use a log-scale grid of choices of λ_m and λ_s (with $\lambda_p = 1.0$ fixed), and for each λ , we tune the RBF/SVM parameters for best classification success in stratified 5-fold cross-validation.

First, we present the most difficult task—the 6-category classification. Figure 14(a) shows the success rate on a grid of λ values. The best classification we achieve is 0.546 (54 of 99) with $\lambda = (0.03, 0.03, 1.0)$. For that λ , Figure 14(b) shows the pairwise distance matrix. In the distance matrix, the pattern of grouping is visually evident—there are small within-group distances compared to some of the between-group differences. Specifically, the subgroups appear to be further from each other for different brain regions but not for different experimental conditions. This partial separation also appears in the MDS plot shown in Figure 14(c), and in the confusion matrix given in Table 2.

TABLE 1
Sample size of each subgroup of the Chen dataset

	Hippocampus CA1	Neocortex Layer V	Total
BDL	15	16	31
BDLHD	15	15	30
Control	18	20	38
Total	48	51	

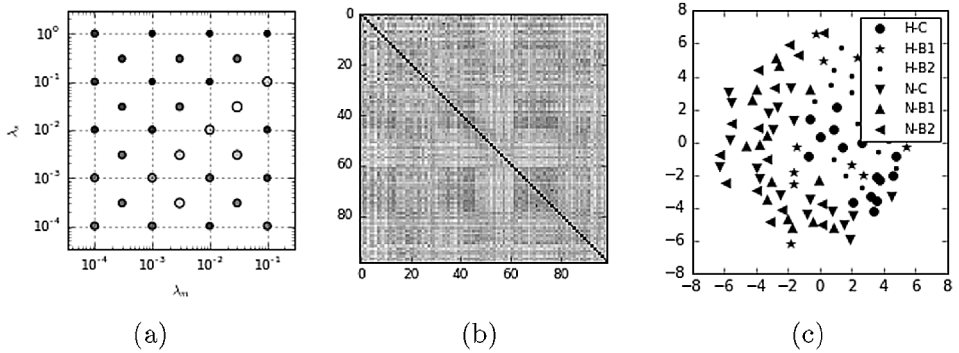


FIG. 14. Combined classification of experimental group and region on the 99 rat dendrites in the Chen dataset. (a) Grid search over distance tuning weights. CV classification accuracy is depicted both by brightness and circle size. Best accuracy is 0.546, achieved with $\lambda = (0.03, 0.03, 1.0)$. (b) Pairwise distance matrix using the best λ . Each class is in a range of consecutive indices: the first 31 are BDL, the next 30 are BDLHD, and the last 38 are control. Within each experimental group, the Hippocampus CA1 neurons are the first part of the range and the Neocortex Layer V neurons are the latter part. (c) MDS plot from the distances in (b). In the legend, Hippocampus and Neocortex are abbreviated as “H” and “N,” respectively. Control, BDL, and BDLHD are abbreviated as “C,” “B1,” and “B2,” respectively.

Next, we try to classify the Chen dataset by brain region only. In the 6-category classification, these two groups appeared to separate, so we expect to get good results when we specifically tune for it. Figure 15(a) shows the CV accuracy over the λ grid. The classification success is high for all the λ we tried, and even reaches perfect classification in some cases. Figure 15(b) shows the pairwise distance matrix for one of these [$\lambda = (0.0003, 0.0003, 1.0)$], and the grouping is clearer than it is in Figure 14(b). The MDS plot in Figure 15(c) shows a clean and unambiguous separation between the two groups.

TABLE 2

Confusion matrix of 6-category classification attempt on the Chen dataset using the proposed method. Columns correspond to predicted class labels in the same order as the rows correspond to predicted labels. Most misclassifications (all but one) are between neurons in the same regions but different experimental group

True Class Label	Predicted class label						
	Control	BDL	BDLHD	H-C	H-B1	H-B2	N-C
CA1	Control	16	1	1	0	0	0
	BDL	5	4	5	0	1	0
	BDLHD	8	3	4	0	0	0
Layer V	Control	0	0	0	14	3	3
	BDL	0	0	0	4	10	2
	BDLHD	0	0	0	8	1	6

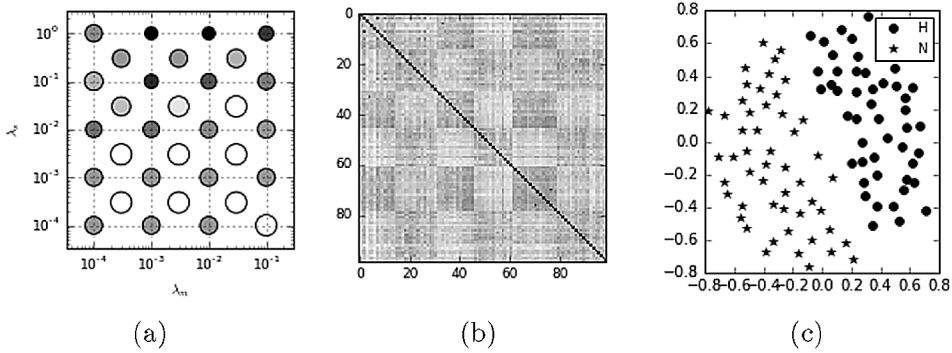


FIG. 15. Classification of brain region on the 99 rat dendrites in the Chen dataset. (a) Grid search over distance tuning weights. CV classification accuracy is depicted both by brightness and circle size. Perfect classification is achieved for several choices of λ shown here. One of these, $\lambda = (0.0003, 0.0003, 1.0)$, is used for the (b)–(c). (b) Pairwise distance matrix using the chosen λ . The index order is the same as in Figure 14(b). (c) MDS plot using the distances in (b).

When tuning the classifier according to all 6 categories or only brain region, we do not see much separation between the three experimental groups. Can we separate them better by explicitly tuning for it? It seems not—we actually do worse. Figure 16(a) shows accuracy over the λ grid and the best accuracy is 0.535 (53 of 99), achieved at $\lambda = (0.03, 0.03, 1.0)$. The corresponding distance matrix and MDS plots are shown in Figure 16(b) and (c), respectively.

3.4. Baseline comparison. For comparison, we attempt the same classification tasks using two other methods: (1) feature extraction and (2) topology-only tree edit distance. The first is a common type of method in neuromorphology. One

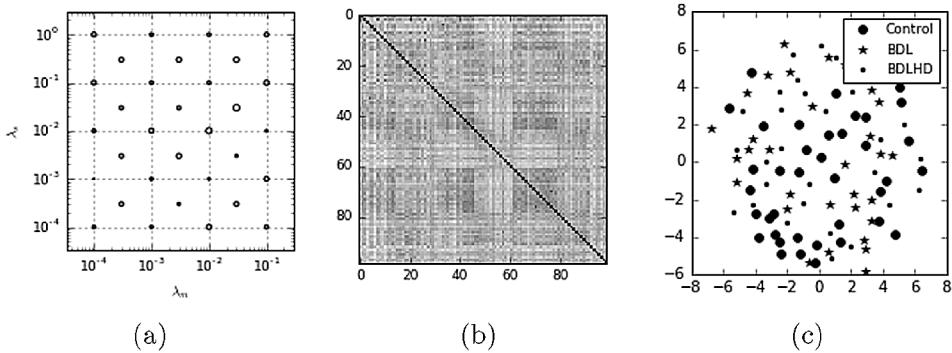


FIG. 16. Classification of brain experimental condition on the 99 rat dendrites in the Chen dataset. (a) Grid search over distance tuning weights. CV classification accuracy is depicted both by brightness and circle size. Best accuracy is 0.535, achieved with $\lambda = (0.03, 0.03, 1.0)$. (b) Pairwise distance matrix using the best λ . The index order is the same as in Figure 14(b). (c) MDS plot from the distances in (b).

TABLE 3

Summary comparison of classification accuracy with Gaussian RBF SVM in Euclidean feature space, topology-only TED metric space, and the proposed metric space of tree shapes

	Wu	Chen (6-class)	Chen (region)	Chen (exp. grp.)
Feature Vector	0.707	0.566	1.000	0.505
TED (topology only)	0.756	0.384	0.859	0.455
Proposed Metric	0.805	0.546	1.000	0.535

constructs a feature vector using high-level morphological measurements of each neuron, and uses those features as input to some standard statistical or machine learning method. This is a somewhat open-ended task because it requires us to choose which features to use and which classifier to apply.

For features, we use data provided directly by NeuroMorpho.org. The site’s listing for each neuron reconstruction includes a list of 21 measurements that summarize its morphological characteristics, for example, soma surface area, number of bifurcations, total length, etc. (the full list can be seen on the website). We obtained the Wu and Chen datasets from that site, so these features are a natural choice since they are already computed and they were presumably chosen by the neuromorphology experts who run the site.

Before considering classification results, we mention an inherent trade-off between our distance and any method that uses this set of features. Our method uses a more complete structure of one neurite tree (for these two datasets we apply it to the apical dendrite), but only that tree. The feature vector includes information from every part of the neuron that is available in the reconstruction, and these two datasets include information from the soma and basal dendrite trees. Thus, the trade-off is that our method is more narrow in scope but more detailed in the information it uses to compare trees.

For the classifier, we use SVM classification with a (Euclidean) Gaussian RBF kernel since that is most analogous to what we used with our proposed shape metric. We standardize the feature data and do a grid search over the RBF/SVM parameters and report the best classification accuracy for the same stratified 5-fold cross-validation. For the tree edit distance, we also use a Gaussian RBF for the same reason. Table 3 lists the accuracy scores for each method on each classification task. Classifying the Chen set by brain region was easiest—both the feature vector and the proposed metric obtained perfect classification. The feature vector method outperformed the proposed metric at 6-category classification on the Chen set, but the proposed method performed best on the remaining two tasks, and by larger margins.

4. Summary. This paper describes a mathematical framework for comparing shapes of neuronal trees. It is based on a metric that is a combination of terms

involving shapes of the main branch, the side branches, and the locations of the side branches. The key idea is to impose an equivalence relation, and a permutation of side branches, that together allow trees with different number of branches to be compared and deformed into each other. This framework is then used to classify trees according to experimental groups, and the results demonstrate the success of this framework using the real datasets.

APPENDIX

Theorem 1 applies to reparameterization of curves. It is a trivial extension to extend this to reparameterization of the indexed collection of curves that make up a tree.

COROLLARY 3 (of Theorem 1). *Let $\mathbf{q}^1, \mathbf{q}^2 \in \mathcal{P}_n$ where each q_k^2 (for $k = 0, \dots, n$) is the SRVF of a piecewise linear curve. There are $\tilde{\mathbf{q}}^1 \in \text{cl}(\mathbf{q}^1 \Gamma^{n+1})$ and $\tilde{\mathbf{q}}^2 \in \text{cl}(\mathbf{q}^2 \Gamma^{n+1})$ such that $d_n(\tilde{\mathbf{q}}^1, \tilde{\mathbf{q}}^2)$ attains the minimum distance between elements of $\text{cl}(\mathbf{q}^1 \Gamma^{n+1})$ and $\text{cl}(\mathbf{q}^2 \Gamma^{n+1})$.*

For Theorem 2, we need to extend this to include rotation as well. To make this clean, we use the following lemma and corollary which give us an intuitive way to write an optimal matching in \mathcal{Q}_n that separates rotation from the closure operator.

LEMMA 2. *Given $\mathbf{q} \in \mathcal{P}_n$, for any $\tilde{\mathbf{q}} \in [\mathbf{q}]$, we can write it as $\tilde{\mathbf{q}} = \tilde{O} \mathbf{q}'$ for some $\tilde{O} \in \text{SO}(3)$ and $\mathbf{q}' \in \text{cl}(\mathbf{q} \Gamma^{n+1})$. That is,*

$$(20) \quad \bigcup_{O \in \text{SO}(3)} O \cdot \text{cl}(\mathbf{q} \Gamma^{n+1}) = \text{cl} \left(\bigcup_{O \in \text{SO}(3)} O \mathbf{q} \Gamma^{n+1} \right) \equiv [\mathbf{q}].$$

In words, the closure of the orbit under the combined action only interacts with the reparameterization component and not the rotation.

PROOF. In terms of inclusion, the set of interest [LHS of equation (20)] is obviously between the orbit and its closure:

$$\bigcup_{O \in \text{SO}(3)} O \mathbf{q} \Gamma^{n+1} \subseteq \bigcup_{O \in \text{SO}(3)} O \cdot \text{cl}(\mathbf{q} \Gamma^{n+1}) \subseteq \text{cl} \left(\bigcup_{O \in \text{SO}(3)} O \mathbf{q} \Gamma^{n+1} \right).$$

[Note the first inclusion implicitly assumes $O \cdot \text{cl}(\mathbf{q} \Gamma^{n+1}) = \text{cl}(O \mathbf{q} \Gamma^{n+1})$, which is true due to isometry of action by $\text{SO}(3)$.] To show equality in equation (20), we need only show that this middle set is closed. Since $\text{SO}(3)$ is compact, its action $\Theta : \text{SO}(3) \times \mathcal{P}_n \rightarrow \mathcal{P}_n$ is a closed map. Thus, the image

$$\Theta(\text{SO}(3) \times \text{cl}(\mathbf{q} \Gamma^{n+1})) = \bigcup_{O \in \text{SO}(3)} O \cdot \text{cl}(\mathbf{q} \Gamma^{n+1})$$

is a closed set, and this completes the proof. \square

COROLLARY 4. *An optimal matching exists between $[q^1], [q^2] \in Q_n$ if and only if $d_n(\tilde{q}^1, \tilde{O}\tilde{q}^2)$ is minimal with respect to $\tilde{q}^1 \in \text{cl}(q^1\Gamma^{n+1}), \tilde{q}^2 \in \text{cl}(q^2\Gamma^{n+1})$, and $\tilde{O} \in \text{SO}(3)$. This $\tilde{q}^1 \in [q^1]$ and $\tilde{O}\tilde{q}^2 \in [q^2]$ is an optimal matching.*

Now we are set up to make a clean, concise proof, in which we show the existence of optimal matching in the form given in Corollary 4.

PROOF OF THEOREM 2 (EXISTENCE OF OPTIMAL MATCHING IN Q_n). Let $q^1, q^2 \in \mathcal{P}_n$ such that q_k^2 (for $k = 0, \dots, n$) is the SRVF of a piecewise linear curve. Let $C = \text{cl}(q^1\Gamma^{n+1}) \times \text{cl}(q^2\Gamma^{n+1})$. Define $g : \text{SO}(3) \times C \rightarrow \mathbb{R}$ by $(O, w^1, w^2) \mapsto d_n(w^1, Ow^2)$.

By Corollary 4, we need only show that g achieves a minimum value on its domain. By Corollary 3, g does achieve a minimum when restricted to a fixed $O \in \text{SO}(3)$. Let $m : \text{SO}(3) \rightarrow \mathbb{R}$ be defined by

$$(21) \quad m(O) = \min_{(w^1, w^2) \in C} g(O, w^1, w^2).$$

We claim m is continuous. For arbitrary $O \in \text{SO}(3)$ and $\varepsilon \in \mathbb{R}^+$, let B be the open ball, $(m(O) - \varepsilon, m(O) + \varepsilon)$. To show m is continuous, it is sufficient to find an open neighborhood, U , of O , such that $m(U) \subset B$.

For any fixed $(\tilde{q}^1, \tilde{q}^2) \in [q^1] \times [q^2]$, let $g_{\tilde{q}^1, \tilde{q}^2} : \text{SO}(3) \rightarrow \mathbb{R}$ be defined by $O \mapsto g(O, \tilde{q}^1, \tilde{q}^2)$. Since the action of $\text{SO}(3)$ is continuous and d_n is continuous, g is clearly continuous, as is $g_{\tilde{q}^1, \tilde{q}^2}$.

For the given O , choose (w^1, w^2) so that $m(O) = d_n(w^1, Ow^2)$, and let $D = g_{w^1, w^2}^{-1}(B)$. Since $g_{w^1, w^2}(O) = m(O) \in B$, and g_{w^1, w^2} is continuous, D is an open neighborhood of O . Also, for any $\tilde{O} \in D$, we have $g_{w^1, w^2}(\tilde{O}) \in B$, which means $m(\tilde{O}) \leq g(\tilde{O}, w^1, w^2) = g_{w^1, w^2}(\tilde{O}) < m(O) + \varepsilon$.

Now let $E = g_{Ow^2, w^2}^{-1}((-\varepsilon, +\varepsilon))$, which is clearly another open neighborhood of O . Let $\tilde{O} \in E$ be arbitrary, and choose $(\tilde{w}^1, \tilde{w}^2) \in C$ such that $m(\tilde{O}) = d_n(\tilde{w}^1, \tilde{O}\tilde{w}^2)$. Due to the isometric action of Γ^{n+1} , we have $d_n(O\tilde{w}^2, \tilde{O}\tilde{w}^2) = d_n(Ow^2, \tilde{O}w^2) = g_{Ow^2, w^2}(\tilde{O}) < \varepsilon$. Then we have

$$\begin{aligned} m(\tilde{O}) + \varepsilon &> m(\tilde{O}) + d_n(O\tilde{w}^2, \tilde{O}\tilde{w}^2) \\ &= d_n(\tilde{w}^1, \tilde{O}\tilde{w}^2) + d_n(O\tilde{w}^2, \tilde{O}\tilde{w}^2) \\ &\geq d_n(\tilde{w}^1, Ow^2) \quad (\text{tri. ineq.}) \\ &\geq m(O) \quad (\text{def. of } m), \end{aligned}$$

or $m(\tilde{O}) > m(O) - \varepsilon$

Now let $U = D \cap E$, which is another open neighborhood of O . If $\tilde{O} \in U$, then $m(O) - \varepsilon < m(\tilde{O}) < m(O) + \varepsilon$, so $m(U) \subset B$, as needed and, therefore, m is continuous.

Since m is a continuous function on a compact domain, it attains a minimum value, and thus g also attains a minimum. Therefore, there exist $\tilde{\mathbf{q}}^1 \in [\mathbf{q}^1]$ and $\tilde{\mathbf{q}}^2 \in [\mathbf{q}^2]$ that attain the minimum distance between the two closed orbits. \square

LEMMA 3. Let $\mathbf{q}^1 \in \mathcal{P}_{n_1}$ and $\mathbf{q}^2 \in \mathcal{P}_{n_2}$ with branches composed of piecewise linear curves, and let $\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2 \in \mathcal{P}_{\tilde{n}}$ be some matching between $[[\mathbf{q}^1]]$ and $[[\mathbf{q}^2]]$. Then there exists a matching, $\mathbf{w}^1, \mathbf{w}^2 \in \mathcal{P}_n$, such that $n \leq \tilde{n}$ and $d_n(\mathbf{w}^1, \mathbf{w}^2)^2 \leq d_{\tilde{n}}(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2)^2$. Furthermore, $\mathbf{w}^1, \mathbf{w}^2$ satisfy the following three properties:

(P1) There are no null branches matched to each other. For each $k = 1, \dots, n$, at most one of w_k^1, w_k^2 is null.

(P2) Null branches do not contribute to the third term of d_n . If w_k^j is null (for some $j \in \{1, 2\}, k \in \{1, \dots, n\}$), then $s_k^1 = s_k^2$.

PROOF OF LEMMA 3. Let $\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2 \in \mathcal{P}_{\tilde{n}}$ be a matching between $[[\mathbf{q}^1]]$ and $[[\mathbf{q}^2]]$. Let $n \leq \tilde{n}$ be the number of indices for which at least one of $\tilde{w}_{i_k}^1, \tilde{w}_{i_k}^2$ is nonnull. Let i_1, i_2, \dots, i_n be those indices.

We construct a matching $\mathbf{w}^1, \mathbf{w}^2 \in \mathcal{P}_n$ from the pieces of $\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2$. First, we keep the same main branches: $w_0^j = \tilde{w}_0^j$ for $j = 1, 2$. Then for each $k = 1, \dots, n$ and $j = 1, 2$, we define the side branch shapes by $w_k^j \leftarrow \tilde{w}_{i_k}^j$. This, with the choice of the i_k , ensures we satisfy (P1). For the side branch locations, we set $s_k^j \leftarrow \tilde{s}_{i_k}^j$ if w_k^j is nonnull. Then we set $s_k^1 \leftarrow s_k^2$ if w_k^1 is null, and likewise set $s_k^2 \leftarrow s_k^1$ if w_k^2 is null. This ensures we satisfy (P2). The only thing left to show is that $d_n(\mathbf{w}^1, \mathbf{w}^2) \leq d_{\tilde{n}}(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2)$.

In the above construction, every nonnull branch in each tree keeps its shape and location from the initial tree, so $\mathbf{w}^j \sim \tilde{\mathbf{w}}^j$. Thus, $\mathbf{w}^1, \mathbf{w}^2$ is another matching between $[[\mathbf{q}^1]]$ and $[[\mathbf{q}^2]]$. Also, for each k , the branch shapes w_k^1, w_k^2 are taken from corresponding branches $\tilde{w}_{i_k}^1, \tilde{w}_{i_k}^2$, so the first two terms of the squared preshape distance are unchanged:

$$\lambda_m \|w_0^1 - w_0^2\|^2 + \lambda_s \sum_{k=1}^n \|w_k^1 - w_k^2\|^2 = \lambda_m \|\tilde{w}_0^1 - \tilde{w}_0^2\|^2 + \lambda_s \sum_{i=1}^{\tilde{n}} \|\tilde{w}_i^1 - \tilde{w}_i^2\|^2.$$

Thus, any change in preshape distance is entirely in the third term. Let $i_{n+1}, \dots, i_{\tilde{n}}$ be the indices for which $\tilde{w}_{i_k}^1$ and $\tilde{w}_{i_k}^2$ are both null. Then the change in squared preshape distance is

$$\begin{aligned} d_n(\mathbf{w}^1, \mathbf{w}^2)^2 - d_{\tilde{n}}(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2)^2 \\ = \lambda_p \sum_{k=1}^n (s_k^1 - s_k^2)^2 - \lambda_p \sum_{i=1}^{\tilde{n}} (\tilde{s}_i^1 - \tilde{s}_i^2)^2 \end{aligned}$$

$$\begin{aligned}
 &= \lambda_p \left(\sum_{k=1}^n (s_k^1 - s_k^2)^2 - \sum_{i=1}^{\tilde{n}} (\tilde{s}_i^1 - \tilde{s}_i^2)^2 \right) \\
 &= \lambda_p \left(\sum_{k=1}^n (s_k^1 - s_k^2)^2 - \sum_{k=1}^n (\tilde{s}_{i_k}^1 - \tilde{s}_{i_k}^2)^2 - \sum_{k=n+1}^{\tilde{n}} (\tilde{s}_{i_k}^1 - \tilde{s}_{i_k}^2)^2 \right) \\
 &= -\lambda_p \left(\sum_{k=1}^n [(\tilde{s}_{i_k}^1 - \tilde{s}_{i_k}^2)^2 - (s_k^1 - s_k^2)^2] + \sum_{k=n+1}^{\tilde{n}} (\tilde{s}_{i_k}^1 - \tilde{s}_{i_k}^2)^2 \right).
 \end{aligned}$$

In the last expression, the terms of the second sum are obviously non-negative. We claim the terms of the first sum are also nonnegative. Let $a_k = (\tilde{s}_{i_k}^1 - \tilde{s}_{i_k}^2)^2 - (s_k^1 - s_k^2)^2$. This k th term corresponds to a pair of branches in the original matching, $\tilde{w}_{i_k}^1, \tilde{w}_{i_k}^2$, where at least one of them is non-null. If both are nonnull, then $s_k^j = \tilde{s}_{i_k}^j$ for $j = 1, 2$, so $a_k = 0$. If only one of the branches is nonnull, then $s_k^1 = s_k^2$, and $a_k \geq 0$.

Therefore, $d_n(\mathbf{w}^1, \mathbf{w}^2) \leq d_{\tilde{n}}(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2)$, and this completes the proof. \square

PROOF OF THEOREM 3 (EXISTENCE OF OPTIMAL MATCHING IN $\tilde{\mathcal{Q}}$). Let $\mathbf{q}^1 \in \mathcal{P}_{n_1}$ and $\mathbf{q}^2 \in \mathcal{P}_{n_2}$ such that q_k^2 (for $k = 0, \dots, n_2$) is the SRVF of a piecewise linear curve.

When considering possible matchings as candidates for an optimal matching, we can restrict our attention to cases which have the properties (P1) and (P2) from Lemma 3. Let $\mathcal{M} \subset \llbracket \mathbf{q}^1 \rrbracket \times \llbracket \mathbf{q}^2 \rrbracket$ be the set of matchings that have those properties. For any such matching, $\mathbf{w}^1, \mathbf{w}^2 \in \mathcal{P}_n$, there are $\tilde{\mathbf{w}}^1 \in \llbracket \mathbf{w}^1 \rrbracket$ and $\tilde{\mathbf{w}}^2 \in \llbracket \mathbf{w}^2 \rrbracket$ such that the $\tilde{\mathbf{w}}^j$ can be obtained from the \mathbf{q}^j by reordering their branch indices in a way consistent with (P1) and (P2). Let \mathcal{W} be the set of such pairs, $(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2)$, obtained by branch reordering alone. Then

$$\mathcal{M} = \bigcup_{(\tilde{\mathbf{w}}^1, \tilde{\mathbf{w}}^2) \in \mathcal{W}} (\mathbf{w}^1, \mathbf{w}^2) \in \llbracket \tilde{\mathbf{w}}^1 \rrbracket \times \llbracket \tilde{\mathbf{w}}^2 \rrbracket.$$

For each $(\mathbf{w}^1, \mathbf{w}^2) \in \mathcal{W}$, there is an optimal matching between $\llbracket \tilde{\mathbf{w}}^1 \rrbracket$ and $\llbracket \tilde{\mathbf{w}}^2 \rrbracket$ (by Theorem 2). \mathcal{W} is clearly finite, so there is some $(\mathbf{w}^1, \mathbf{w}^2) \in \mathcal{M}$ that minimizes the preshape distance, and that minimizing pair is an optimal matching between $\llbracket \mathbf{q}^1 \rrbracket$ and $\llbracket \mathbf{q}^2 \rrbracket$. \square

Acknowledgments. The authors would like to thank Xavier Descombes and his collaborators and students at INRIA Sophia Antipolis and IBV for supplying one of the datasets, and Neuromorpho.org [Ascoli, Donohue and Halavi (2007)] for providing the other two.

REFERENCES

- ANDERSSON-ENGELS, S., AF KLINTEBERG, C., SVANBERG, K. and SVANBERG, S. (1997). In vivo fluorescence imaging for tissue diagnostics. *Phys. Med. Biol.* **42** 815–824.
- ASCOLI, G. A., DONOHUE, D. E. and HALAVI, M. (2007). NeuroMorpho.Org: A central resource for neuronal morphologies. *J. Neurosci.* **27** 9247–9251.
- AYDIN, B., PATAKI, G., WANG, H., BULLITT, E. and MARRON, J. S. (2009). A principal component analysis for trees. *Ann. Appl. Stat.* **3** 1597–1615. [MR2752149](#)
- AYDIN, B., PATAKI, G., WANG, H., LADHA, A. and BULLITT, E. (2011). Visualizing the structure of large trees. *Electron. J. Stat.* **5** 405–420. [MR2802049](#)
- BASSELL, G. J. and WARREN, S. T. (2008). Fragile X syndrome: Loss of local mRNA regulation alters synaptic development and function. *Neuron* **60** 201–214.
- BILLERA, L. J., HOLMES, S. P. and VOGTMANN, K. (2001). Geometry of the space of phylogenetic trees. *Adv. in Appl. Math.* **27** 733–767. [MR1867931](#)
- BRUVERIS, M. (2015). Optimal reparametrizations in the square-root velocity framework. *SIAM Journal on Mathematical Analysis* **48** 4335–4354.
- CHAN-PALAY, V. and ASAN, E. (1989). 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. *The Journal of Comparative Neurology* **287** 373–392.
- CHEN, J.-R., WANG, B.-N., TSENG, G.-F., WANG, Y.-J., HUANG, Y.-S. and WANG, T.-J. (2014). Morphological changes of cortical pyramidal neurons in hepatic encephalopathy. *BMC Neuroscience* **15** 15.
- COLEMAN, P. D. and FLOOD, D. G. (1987). Neuron numbers and dendritic extent in normal aging and Alzheimer’s disease. *Neurobiol. Aging* **8** 521–545.
- CUNTZ, H., FORSTNER, F., HAAG, J. and BORST, A. (2008). The morphological identity of insect dendrites. *PLoS Comput. Biol.* **4** e1000251.
- DRYDEN, I. L. and MARDIA, K. V. (1998). *Statistical Shape Analysis*. Wiley, Chichester. [MR1646114](#)
- DUNCAN, A., SRIVASTAVA, A., DESCOMBES, X. and KLASSEN, E. (2015). Geometric analysis of axonal tree structures. In *DIFF-CV: Differential Geometric Techniques in Computer Vision*.
- ENGLE, E. C. (2008). Human genetic disorders of axon guidance. *Neuron* **60** 201–214.
- FERAGEN, A. (2012). Complexity of computing distances between geometric trees. In *Structural, Syntactic, and Statistical Pattern Recognition: Joint IAPR International Workshop, SSPR&SPR 2012, Hiroshima, Japan, November 7–9, 2012. Proceedings* 89–97. Springer, Berlin.
- FERAGEN, A., LAUZE, F. and HAUBERG, S. (2015). Geodesic exponential kernels: When curvature and linearity conflict. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- FERAGEN, A., HAUBERG, S., NIELSEN, M. and LAUZE, F. (2011). Means in spaces of tree-like shapes. In *Computer Vision (ICCV), 2011 IEEE International Conference on* 736–746.
- FERAGEN, A., OWEN, M., PETERSEN, J., WILLE, M. M., THOMSEN, L. H., DIRKSEN, A. and DE BRUIJNE, M. (2013a). Tree-space statistics and approximations for large-scale analysis of anatomical trees. In *Information Processing in Medical Imaging: 23rd International Conference, IPMI 2013, Asilomar, CA, USA, June 28–July 3, 2013. Proceedings* 74–85. Springer, Berlin.
- FERAGEN, A., LO, P., DE BRUIJNE, M., NIELSEN, M. and LAUZE, F. (2013b). Toward a theory of statistical tree-shape analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 2008–2021.
- FERAGEN, A., PETERSEN, J., OWEN, M., LO, P., THOMSEN, L. H., WILLE, M. M., DIRKSEN, A. and DE BRUIJNE, M. (2015). Geodesic atlas-based labeling of anatomical trees: Application and evaluation on airways extracted from CT. *IEEE Trans. Med. Imag.* **34** 1212–1226.
- GIBSON, D. A. and MA, L. (2011). Developmental regulation of axon branching in the vertebrate nervous system. *Development* **138** 183–195.

- HALAVI, M., HAMILTON, K. A., PAREKH, R. and ASCOLI, G. A. (2012). Digital reconstructions of neuronal morphology: Three decades of research trends. *Frontiers in Neuroscience* **6**.
- HEUMAN, H. and WITTUM, G. (2009). The tree-edit-distance, a measure for quantifying neuronal morphology. *Neuroinformatics* **7** 179–190.
- HIROKAWA, N., NIWA, S. and TANAKA, Y. (2010). Molecular motors in neurons: Transport mechanisms and roles in brain function, development, and disease. *Neuron* **68** 610–638.
- JONKER, R. and VOLGENANT, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* **38** 325–340. [MR0902027](#)
- JOSHI, S. H., KLASSEN, E., SRIVASTAVA, A. and JERMYN, I. (2007). A novel representation for Riemannian analysis of elastic curves in \mathbb{R}^n . In *2007 IEEE Conference on Computer Vision and Pattern Recognition* 1–7.
- KABASO, D., COSKREN, P., HENRY, B., HOF, P. and WEARNE, S. (2009). The electrotonic structure of pyramidal neurons contributing to prefrontal cortical circuits in macaque monkeys is significantly altered in aging. *Cerebral Cortex* **19** 2248–2268.
- KENDALL, D. G., BARDEN, D., CARNE, T. K. and LE, H. (1999). *Shape and Shape Theory*. Wiley, Chichester. [MR1891212](#)
- KURTEK, S., SRIVASTAVA, A., KLASSEN, E. and DING, Z. (2012). Statistical modeling of curves using shapes and related features. *J. Amer. Statist. Assoc.* **107** 1152–1165. [MR3010902](#)
- LAHIRI, S., ROBINSON, D. and KLASSEN, E. (2015). Precise matching of PL curves in \mathbb{R}^N in the square root velocity framework. *Geom. Imaging Comput.* **2** 133–186. [MR3501512](#)
- LEDDEROSE, J., SENCION, L., SALGADO, H., ARIAS-CARRION, O. and TREVINO, M. (2014). A software tool for the analysis of neuronal morphology data. *Int. Archive Medicine* **7** 1–9.
- LIU, W., SRIVASTAVA, A. and KLASSEN, E. (2008). Joint shape and texture analysis of objects boundaries in images using a Riemannian approach. In *Asilomar Conference on Signals, Systems, and Computers*.
- LIU, W., SRIVASTAVA, A. and ZHANG, J. (2011). A mathematical framework for protein structure comparison. *PLoS Comput. Biol.* **7** e1001075, 10. [MR2788145](#)
- MEDIONI, C., RAMIALISON, M., EPHRUSSI, A. and BESSE, F. (2014). Imp promotes axonal remodeling by regulating profilin mRNA during brain development. *Curr. Biol.* **24** 793–800.
- MIO, W. and SRIVASTAVA, A. (2004). Elastic-string models for representation and analysis of planar shapes. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.* **2** II–10–II–15.
- NTZIACHRISTOS, V. (2006). Fluorescence molecular imaging. *Annu Rev Biomed Eng* **8** 1–33.
- SCHOENBERG, I. J. (1938). Metric spaces and positive definite functions. *Trans. Amer. Math. Soc.* **44** 522–536. [MR1501980](#)
- SELKOW, S. M. (1977). The tree-to-tree editing problem. *Inform. Process. Lett.* **6** 184–186. [MR0458995](#)
- SRIVASTAVA, A. and KLASSEN, E. P. (2016). *Functional and Shape Data Analysis*. Springer, New York.
- SRIVASTAVA, A., KLASSEN, E., JOSHI, S. H. and JERMYN, I. H. (2011). Shape analysis of elastic curves in Euclidean spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **33** 1415–1428.
- SUO, L., LU, H., YING, G., CAPECCHI, M. R. and WU, Q. (2012). Protocadherin clusters and cell adhesion kinase regulate dendrite complexity through Rho GTPase. *Journal of Molecular Cell Biology* **4** 362.
- TAI, K. C. (1979). The tree-to-tree correction problem. *J. Assoc. Comput. Mach.* **26** 422–433. [MR0535263](#)
- WANG, H. and MARRON, J. S. (2007). Object oriented data analysis: Sets of trees. *Ann. Statist.* **35** 1849–1873. [MR2363955](#)
- WEST, M. J., COLEMAN, P. D., FLOOD, D. G. and TRONCOSO, J. C. (1994). Differences in the pattern of hippocampal neuronal loss in normal ageing and Alzheimer’s disease. *Lancet* **344** 769–772.

- WHITEHOUSE, P. J., PRICE, D. L., STRUBLE, R. G., CLARK, A. W., COYLE, J. T. and DELON, M. R. (1982). Alzheimer's disease and senile dementia: Loss of neurons in the basal forebrain. *Science* **215** 1237–1239.
- WU, C. H. et al. (2012). Mutations in the proflin 1 gene cause familial amyotrophic lateral sclerosis. *Nature* **488** 499–503.
- ZHANG, K. (1996). A constrained edit distance between unordered labeled trees. *Algorithmica* **15** 205–222. [MR1368250](#)

A. DUNCAN
 A. SRIVASTAVA
 DEPARTMENT OF STATISTICS, MAILCODE 4330
 FLORIDA STATE UNIVERSITY
 ROOM 210D, BUILDING OSB
 TALLAHASSEE, FLORIDA 32306
 USA
 E-MAIL: a.duncan@stat.fsu.edu

E. KLASSEN
 DEPARTMENT OF MATHEMATICS
 FLORIDA STATE UNIVERSITY
 ROOM 114, LOVE BUILDING
 TALLAHASSEE, FLORIDA 32306
 USA