Cylindrical Shape Decomposition for 3D Segmentation of Tubular Objects

ALI ABDOLLAHZADEH[ID], **ALEJANDRA SIERRA**[ID], **AND JUSSI TOHKA**[ID]

A. I. Virtanen Institute for Molecular Sciences, University of Eastern Finland, 70211 Kuopio, Finland

Corresponding author: Ali Abdollahzadeh (ali.abdollahzadeh@uef.fi)

ABSTRACT We develop a cylindrical shape decomposition (CSD) algorithm to decompose an object, a union of several tubular structures, into its semantic components. We decompose the object using its curve skeleton and restricted translational sweeps. For that, CSD partitions the curve skeleton into maximal-length sub-skeletons over an orientation cost, each sub-skeleton corresponds to a semantic component. To find the intersection of the tubular components, CSD translationally sweeps the object in decomposition intervals to identify critical points at which the shape of the object changes substantially. CSD cuts the object at critical points and assigns the same label to parts along the same sub-skeleton, thereby constructing a semantic component. The proposed method further reconstructs the acquired semantic components at the intersection of object parts using generalized cylinders. We apply CSD for segmenting axons in large 3D electron microscopy images and decomposing vascular networks and synthetic objects. We show that our proposal is robust to severe surface noise and outperforms state-of-the-art decomposition techniques in its applications.

INDEX TERMS Cylindrical decomposition, electron microscopy, generalized cylinder, image segmentation, skeleton decomposition, tubular object decomposition.

I. INTRODUCTION

Shape decomposition is a fundamental problem in geometry processing where an arbitrary object is regarded as an arrangement of simple primitives [1], [2] or semantic components [3], [4]. Applications of shape decomposition include disciplines such as object recognition and retrieval [5], [6], shape reconstruction [7], shape clustering [8], or modeling [9].

Our motivation for studying shape decomposition comes from biomedical image segmentation. Advanced biomedical imaging techniques, such as 3D electron microscopy, generate large image volumes whose size can range from a gigabyte to hundreds of terabytes [10]–[12]. Segmentation of such image volumes generally favors bottom-up strategies, where the image is first over-segmented into supervoxels, then supervoxels are merged subsequently [13]–[15]. This strategy is error-prone because both the over-segmentation and subsequent merge are subjected to greedy optimization

as opposed to optimizing a global objective. Our idea is instead to approach the segmentation problem based on a top-down strategy, where under-segmentation is followed by subsequent split using *a priori* knowledge of objects to be segmented; in our case, tubularity of neuronal processes or blood vessels. This strategy divides a large image volume into sub-domains whose geometry/topology can be analyzed based on a global objective, independently and in parallel, but necessitates the development of a fast and robust shape decomposition technique capable of processing thousands of tubular structures in a reasonable time.

This article develops a novel decomposition algorithm called cylindrical shape decomposition (CSD), decomposing big voxel-based tubular objects in large image volumes. We demonstrate the CSD's application in segmenting tubular structures, as the split operation of a top-down strategy, and its application in decomposing general synthetic objects.

A. RELATED WORK

We categorize shape decomposition techniques in the literature into three categories: 1) representing an object in terms

---

The associate editor coordinating the review of this manuscript and approving it for publication was Kumaradevan Punithakumar[ID].

of geometrically homogeneous and simple primitives, such as ellipsoids, convex components, or generalized cylinders [1], [2], [7], [16]–[20]; 2) decomposing an object into its semantic components using object skeleton or Reeb graph [3], [4], [21], [22]; and 3) learning-based decomposition methods [23]–[26].

Primitives are homogeneous components with a compact representation and efficient computation. Examples of primitives with a simple parametric representation include ellipsoids [16] and straight cylinders [17]. This class of primitives with a simple parametric representation is typically applied in description simplification of complex geometrical models. Therefore, higher-level primitives such as tubular primitives [18], convex components [1], [19], generalized cylinders [2], and generalized sweep components [7], [20] were proposed for trading-off the representation simplicity for the generality. For example, tubular primitives in Plumber [18] are constructed applying a seeded region growing with a heuristic set of sphere positions and radii. The Plumber does not return a complete decomposition of objects but extracts ideal tubular components [18]. Convexity-based methods are another interesting class of high-level primitive-based decomposition techniques, developed based on the human tendency to divide an object into parts around concave regions [27]. An exact decomposition of a shape into convex components is costly and too strict for decomposition because such methods can generate many small parts. Therefore, [19] and [1] apply weakly convex components, which are derived from analyzing the pairs of points in the shape visible to each other, obtaining an approximate convex decomposition (ACD). An alternative to the convex decomposition is generalized cylinder decomposition (GCD), quantifying cylindricity. Reference [2] introduces a quantitative measure for the cylindricity following the minimum description length principle [28] as a measure of the skeleton straightness and the variation among the profiles. In this method, the global objective for merging local generalized cylinders is to minimize the cylindricity. The approximate convexes and generalized cylinders are effective high-level primitives, where the approximate convex method generates smoother cuts between parts, and the generalized cylinders suit better the decomposition of tubular objects. The generalized cylinders method is a computationally expensive technique, e.g., [22] reported approximately 30 minutes for a single femur decomposition. Another set of high-level primitive-based decomposition techniques is based on cross-sectional sweeping. These methods are computationally less demanding as compared to the convexity and generalized cylinder methods. The sweeping algorithms analyze object cross-sections and generate homogeneous sweeping components. For example, in [20], the object is swept along its curve skeleton in search of critical points, where the object geometry/topology changes substantially. This method uses the variation of the perimeter of consecutive cross-sections as the homogeneity measure, which is sensitive to the surface noise and prone to over-segmentation. The method in [7] generates local prominent cross-sections from a set of initial seed points. This method is semi-automated, requiring user interactions to adjust the density of cross-sections in different object regions and avoid creating prominent cross-sections in regions with no sweep evidence.

To decompose an object into its semantic components, the object curve skeleton or Reeb graph can be used. Both concepts are object descriptors able to guide a decomposition: the curve skeleton is a 1D representation of a 3D object [29], encoding its topology and geometry; the Reeb graph tracks topology changes in level sets of a scalar function [29]. Reference [21] extracts object curve skeleton based on a collapse measure, i.e., a measure of importance, and subsequently provide an object decomposition by defining skeleton-to-surface mapping based on the shortest geodesics. Reference [3] extracts the curve skeleton, applying an implicit Laplacian smoothing with global positional constraints, preserving the mesh connectivity and its key features. Reference [3] provides an object decomposition with an approximate measure of thickness about extracted curve skeletons. The tubular decomposition in [22] aims to be as close as possible to a Voronoi partitioning, having skeleton branches as sites, while satisfying structural constraints that ensure each decomposition element is a tube-like shape. These skeleton-based decomposition methods [3], [21], [22] are applied to segment synthetic objects into functional parts, which is not the case for decomposing an under-segmentation error in objects acquired from biomedical imaging datasets. In [4], the decomposition of a 3D mesh is a two-step approach accounting for the Reeb graph construction and refinement: the Reeb graph captures the surface topology and protrusions, and the refinement step uses curvature and adjacency information on the graph critical points for fine localization of part boundaries. This approach does not provide smooth boundary cuts between parts, requiring the internal energy function to control the smoothness of boundaries. In the context of skeletonization, it is worth reviewing the L1-medial skeletonization [30] and rotational symmetry axis (ROSA) [31] techniques. The L1-medial skeletonization employs localized L1-medians to construct a skeleton. This method uses a weighting function with a supporting radius that defines the size of the local neighborhood; gradually increasing the supporting radius yields a clean and well-connected skeleton. ROSA defines a curve skeleton as a generalized rotational symmetry axis of a shape. The position of a skeleton point in a local set of points is computed by minimizing the sum of the projected distances to the normal extensions of the data points. The L1-medial and ROSA skeletonization techniques may form cycles when two object parts are close to each other, where the skeleton is to be acyclic. These two skeletonization methods deal with incomplete point clouds, but the skeleton centeredness within the objects is not guaranteed.

Learning-based methods are an important class of shape decomposition techniques, from early statistical modeling methods [23] to recent deep neural network techniques [24]–[26]. The objective function of a
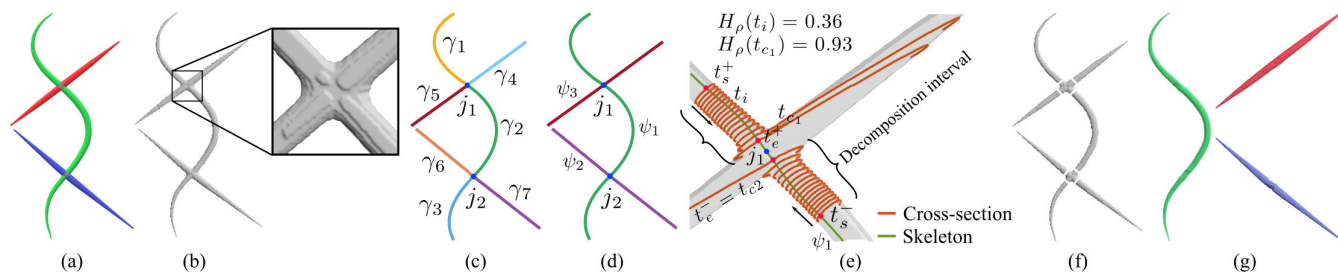
FIGURE 1. Outline of the CSD algorithm. (a) An object is a union of several tubular components. The tubular components are color-coded. (b) A 800 × 400 × 70 voxel-based representation of the object. Intersections of the tubular components are magnified. (c) The curve skeleton of the synthetic object in (b) is the union of all skeleton branches. Skeleton branches are color-coded and denoted as γ. We define a junction-point j as such a point that skeleton branches connect. Junction-points are shown as blue filled-circles. (d) The curve skeleton of the object is partitioned into maximal-length sub-skeletons ψ over a local orientation cost. The sub-skeletons are color-coded. (e) On a sub-skeleton ψ and in the proximity of a junction-point $j \in \psi$, we define two decomposition intervals. The boundaries of decomposition intervals are shown with red filled-circles. The object is swept along ψ and towards the joint j to find a critical point in each interval. At a critical point, the normalized Hausdorff distance H_ρ between a cross-sectional contour and the mean of visited cross-sectional contours exceeds θ_H. Sweeping directions are shown with arrows. (f) We cut the object at critical points to obtain object parts. (g) The object parts along the same sub-skeleton are assigned the same label to construct a semantic-component. The semantic-components are further reconstructed between their comprising object-parts using generalized cylinders. The synthetic object in (a) comprises seven object-parts, and our algorithm decomposes it into three semantic components.

learning-based method is learned from a collection of labeled training objects. Learning-based decomposition methods have demonstrated impressive results, often producing segmentation and labeling comparable to those produced by humans. However, these techniques crucially depend on large training datasets and are often impaired when the objects to be decomposed deviate substantially from the training material.

B. OUTLINE OF THE CSD ALGORITHM AND CONTRIBUTIONS

The main idea of the CSD algorithm is to guide the decomposition using the object curve skeleton and cut the object by restricted translational sweeps. Fig. 1a shows an under-segmented tubular object as a union of three tubular components. CSD begins with extracting the object curve skeleton (Fig. 1c) and partitioning the skeleton into maximal-length sub-skeletons over an orientation cost function (Fig. 1d). Each sub-skeleton corresponds to a semantic tubular component. To identify intersections of the semantic components, CSD translationally sweeps the object along sub-skeletons, searching for critical points where the object cross-section changes substantially (Fig. 1e). A translational sweep is restricted in decomposition intervals; in the proximity of junction-points where sub-skeletons intersect. The object is cut at critical points to obtain object parts (Fig. 1f). A semantic component is further reconstructed at intersections, using generalized cylinders (Fig. 1g).

The CSD algorithm possesses several advantages over previous shape decomposition techniques. Unlike semi-automated methods in [7] and [3], CSD is a fully automatic algorithm and requires no manual interventions; the decomposition is guided using an algorithm that partitions the object skeleton curve into distinct maximal-length straight sub-skeletons. Compared to primitive-based methods in [1], [2], [16], [17], or skeleton-to-surface mapping in [4], [21], our method identifies the intersection of the object parts and

defines smooth boundary cuts between them. Compared to [20], our method is intrinsically more robust in defining critical points in the presence of noise because we measure cross-sectional changes using the mean close curve of traversed cross-sections and modified Hausdorff distance. In [2], generating a generalized cylinder requires iterative operations, yet many such primitives are required to cover the object; henceforth, these primitives must be merged to satisfy a global objective. Such methods are computationally expensive, not suitable for the decomposition of big voxel-based objects in large image volumes. Unlike [3], [21], [22], and [4], which apply a one-to-one assignment between skeleton branches and object parts, we propose to merge skeleton branches belonging to the same semantic part. Unlike learning-based techniques [23]–[26], our proposal does not rely on training and thus generalizes to the variation of tubular objects extracted from medical images, where it attains consistent quality without the need for additional training datasets. In comparison to the L1-medial and ROSA skeletonization techniques, we use a distance-based skeletonization approach, which correctly stays at the center of the object, even when the object parts are adjacent.

In the experimental section of this article, we demonstrate the application of CSD in the segmentation of large electron microscopy volumes of myelinated axons. We also demonstrate the CSD decomposition of vascular networks and synthetic objects. Moreover, we compare CSD to other state-of-the-art decomposition techniques (ACD [1] and GCD [2]), and our skeletonization technique to well-known skeletonization approaches (L1-medial and ROSA). We also evaluate the effect of surface noise on decomposition results and assess a methodology to reduce the CSD computation-time.

II. PRELIMINARIES

This section defines the core concepts used in the paper as there are no generally accepted definitions for most of them.

A. OBJECT

An object $\Omega \subset \mathbb{R}^3$ is a nonempty bounded open set. We assume that its boundary $\partial \Omega$ is homeomorphic to a 2-sphere. For a discrete object, which results from foreground segmentation, we define a 3D binary image as $I : X \subset \mathbb{Z}^3 \rightarrow \{0, 1\}$, and a segmented object $\Omega := \{x \in X : I(x) = 1\}$, where X is the image domain. Throughout the paper, Ω, $\partial \Omega$, and x are in $\mathbb{R}^3$ unless defined otherwise.

B. CURVE SKELETON

Given Ω and $\partial \Omega$, the curve skeleton $\Upsilon \subset \Omega$ is defined as a locus of centers of maximal inscribed balls [32]. A ball $B(x, r)$ centered at $x \in \Omega$ with radius r is maximally inscribed if its surface touches $\partial \Omega$ in at least two distinct points. Formally, B is a maximal inscribed ball in Ω if $\forall B'$, $B \subseteq B' \subseteq \Omega \Rightarrow B' = B$.

C. CURVE SKELETON POINT TYPE

We distinguish three types of points on the curve skeleton of an object: 1) regular-points that have exactly two neighbor points on the skeleton, 2) end-points that have exactly one neighbor point on the skeleton, and 3) junction-points that have three or more neighbor points on the skeleton [29]. We denote the collection of junction-points as J where $j \in J$ and the collection of end-points as O where $o \in O$.

D. SKELETON BRANCH

Removing junction-points J from the curve skeleton Υ results in disconnected simple curves, known as skeleton branches. The collection of skeleton branches is denoted as Γ, and a skeleton branch is $\gamma \in \Gamma$. For $\gamma(t) : [0, 1] \rightarrow \mathbb{R}^3$, its arc-length is written as $l = \int_0^1 |\dot{\gamma}(t)| \, dt$ with the convention $\dot{\gamma}(t) := \dfrac{d}{dt} \gamma(t)$.

E. SKELETON GRAPH

The topology of curve-skeleton Υ can be represented as a connected acyclic undirected graph (i.e., a tree) $G_\Upsilon = (V, E, L)$. There is a one-to-one map between skeleton branches in Γ and edges in E and a one-to-one map between the union of end-points and junction-points $(O \cup J)$ and vertices in V. This means that for each branch $\gamma \in \Gamma$ we associate exactly one edge in e in G_Υ. $L \subset \mathbb{R}^+$ is the set of edge lengths. The length of an edge is the arc-length of its associated skeleton branch.

F. WALK, PATH

A walk is a finite or infinite sequence of edges which joins a sequence of vertices. A finite walk is a sequence of edges $W = \{e_1, e_2, \dots, e_{n'}\}$ for which there is a sequence of vertices $\{v_0, v_1, \dots, v_{n'}\}$ such that $e_i = v_{i-1}v_i$ for $i = 1, \dots, n'$. The vertex sequence of the walk is $(v_0, v_1, \dots, v_{n'})$. A path is a walk in which all vertices are distinct.

Sub-skeleton is a path in the curve skeleton domain. If $W = \{e_1, e_2, \dots, e_{n'}\}$ is a path in the skeleton graph, and $\{\gamma_1, \gamma_2, \dots, \gamma_{n'}\}$ are corresponding skeleton branches, then $\psi = \cup_i \gamma_i \subseteq \Upsilon$ is a sub-skeleton.

G. CRITICAL POINT

A point on a sub-skeleton at which the cross-sectional contour of the object changes substantially. We provide a formal definition in section V-B.

H. CUT

A closed simple curve $C \subset \partial \Omega$ is a cutting-curve if $\partial \Omega \setminus C$ is not connected. Cut means removal of a cutting-curve from the surface.

III. OUTLINE OF THE CSD ALGORITHM

The outline of the CSD algorithm is shown in Fig. 1, and it is as follows:

1) define the curve skeleton of a given object (Fig. 1c, section IV-A);
2) partition the curve skeleton of the object into sub-skeletons (Fig. 1d, section IV-B);
3) define decomposition intervals to restrict the object sweep (Fig. 1e, section V-A);
4) sweep the object to find critical points and cut the object at critical points (Fig. 1e and Fig. 1f, section V-B);
5) reconstruct the object between parts that have the same label using generalized cylinders (Fig. 1g, section VI).

The CSD algorithm is designed for genus zeros objects.

IV. SKELETON PARTITIONING

We use the curve skeleton of an object to drive the decomposition. For that, we partition the skeleton graph into several distinct paths union of which covers the skeleton graph. The partitioning of the skeleton graph, by extension, partitions the curve skeleton into sub-skeletons. Each sub-skeleton corresponds to exactly one semantic object component.

A. CURVE SKELETON

To determine the curve skeleton of an object Ω with sub-voxel precision, we apply a method from [33] and [34]. The algorithm initiates by determining a point $x^* \in \Omega$ with the biggest distance from the object surface $\partial \Omega$ inside the object domain. This point is used to determine a skeleton branch $\gamma(t) : [0, 1] \rightarrow \mathbb{R}^3$, starting at x_s, the furthest geodesic point from x^* in Ω, and ending at x^*. A cost function F is defined to enforce the path to run in the middle of Ω, where F should increase if the path moves away from the center. To determine F, we find the distance field $D(x)$ from $\partial \Omega$, and assign $F = 1 - \left(\dfrac{D(x)}{D(x^*)} \right)^2$. The distance field $D(x)$ is determined by solving an Eikonal equation on the object domain Ω using the fast marching method [35]. Starting at x_s, the skeleton branch γ is traced by a back-tracking procedure on F to reach x^*, written as

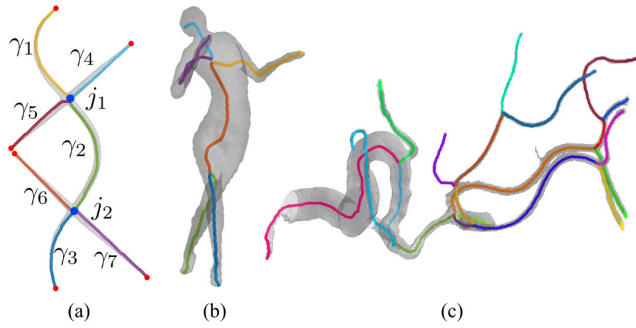$$\gamma = \arg\min_P \int_{x_s}^{x^*} F(P(t)) \, dt, \tag{1}$$

FIGURE 2. The curve skeleton of an object is the union of all skeleton branches. (a) The skeleton of the synthetic object, size: 800 × 400 × 70 voxels, seven branches. The blue filled-circles show junction-points, and the red filled-circles show end-points. The skeleton graph of this object is $G_\Upsilon(V, E, L)$, where $E = \{e_1, \ldots, e_7\}$, $V = \{v_0, \ldots, v_7\}$, and $L = \{l_1, \ldots, l_7\}$. (b) The skeleton of a synthetic object, size: 128 × 128 × 128 voxels, six branches. (c) The skeleton of a vascular network, size: 256 × 256 × 256 voxels, 20 branches. Skeleton branches are color-coded.

FIGURE 3. Partitioning the skeleton graph of the synthetic tubular object. E comprises seven edges and is partitioned into three paths: $W_1 = \{e_1, e_2, e_3\}$, $W_2 = \{e_4, e_5\}$, and $W_3 = \{e_6, e_7\}$. We determine W_1 starting from the longest edge in E denoted as e^* towards its incident vertices. At each vertex, we traverse the edge with minimum orientation cost. Appending new edges terminates when a leaf vertex is visited, or the angle between two successive edges is smaller than θ_c. We subtract W_1 from E, when W_1 is determined (see Algorithm 1). The blue filled-circles show vertices in G_Υ. The edges are color-coded with full-lines. At vertices, arrows show where to traverse next when standing on e^*. The grey dash-lines show the previously calculated paths.

where t traces the path P. We use the Euler scheme for the back-tracking procedure, which solves the ordinary differential equation with a sub-voxel accuracy. This process is repeated to determine further branches that form the curve skeleton of the object. But rather than using the single point x^* as the starting point for the fast marching method, all points in the previously calculated branches are used as starting points. We propagate a new wave from the starting points with the speed F to update x_s. The point x_s is now the furthest point from the current state of the curve skeleton and the starting point of the new branch. Applying a back-tracking algorithm from the updated x_s defines the new skeleton branch. Fig. 2 shows the skeletons of two synthetic objects and a vascular network.

B. SKELETON GRAPH DECOMPOSITION

Several skeleton branches are often required to represent one semantic component of an object, and therefore detecting skeleton branches is not sufficient for a semantic decomposition. An example is shown in Fig. 2a, where the union of three skeleton branches γ_1, γ_2, and γ_3 is required to represent one tubular component. To formalize what constitutes a semantic decomposition, we consider connectivity, length, and local orientation, to unify skeleton branches. We propose an algorithm for traversing the graph representation of the curve skeleton $G_\Upsilon(V, E, L)$, decomposing G_Υ into distinct paths, each corresponds to a semantic component. The algorithm starts at the root edge and explores as far as possible along edges, which provide the optimal choice at each stage.

We partition the skeleton graph of the object into several distinct paths union of which covers the set of graph edges. Formally, we partition $G_\Upsilon(V, E, L)$ into m paths W_i, $i = 1, \ldots, m$ so that $\cup_i W_i = E$ and $W_i \cap W_k = \emptyset \ \forall i, k = 1, \ldots, m, \ i \neq k$. To determine the paths, we require four conditions: 1) the path contains the longest edge not associated to any other path, 2) the path has the maximum number
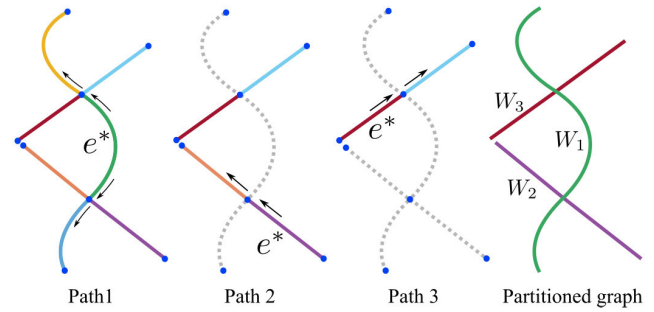
of edges, 3) the associated angle between two successive edges is bigger than θ_c, and 4) the path locally minimize an orientations cost. Denoting two successive edges in a path as e_s and e_{s+1}, the edge e_{s+1} has the maximum angle compared to e_s among the set of connected edges to e_s. The angle between two edges e_s and e_{s+1} is the angle between the line segments connecting endpoints of the skeleton branches associated with edge e_s and e_{s+1}, and it lies in range $[0, \pi]$. We used Algorithm 1 to determine the m distinct paths on G_Υ. Fig. 3 shows skeleton graph decomposition of the synthetic object $n = 7$ into three paths $m = 3$. Each path is equivalent to a sub-skeleton.

V. CYLINDRICAL DECOMPOSITION

In this section, we propose a method to decompose an object into parts and intersections by cutting the object at critical points. To determine critical points, we sweep the object along sub-skeletons in decomposition intervals to find locations where the object geometry changes substantially (see Fig. 6).

A. DECOMPOSITION INTERVAL

We restrict the sweep of the object along each sub-skeleton to decomposition intervals in the proximity of a junction-point j on sub-skeleton ψ, as illustrated in Fig. 4. It is convenient to work with parametrized sub-skeleton $\psi(t) : [0, 1] \to \mathbb{R}^3$. We define two decomposition intervals $[t_s^+, t_e^+]$ and $[t_e^-, t_s^-]$ for each junction-point as in Fig. 4a. To determine the boundaries of a decomposition interval, we define an upper threshold r_s and a lower threshold r_e. We specify r_s and r_e based on the radius of the maximal inscribed ball at t_j and two factors $\alpha_s \geq 1$ and $\alpha_e \geq 0$ where $\alpha_s \geq \alpha_e$, as $r_s = \alpha_s \times r$ and $r_e = \alpha_e \times r$.

To determine the thresholds, we use the signed arc-length from j. Define t_j so that $j = \psi(t_j)$. Then t_s^+ (t_s^-) is

Input : $G_{\Upsilon} = (V, E, L); \theta_c$.
Output: Collection of distinct paths Λ.
$\Lambda \leftarrow \emptyset$
while $E \neq \emptyset$ **do**
    $W \leftarrow \emptyset; e^* \leftarrow$ longest $e \in E$
    $V^* \leftarrow$ { incident vertices to e^* }
    $W \leftarrow W \cup \{e^*\}$
    forall $\upsilon \in V^*$ **do**
        $e^{ref} \leftarrow e^*$
        $\upsilon^{next} \leftarrow \upsilon$
        while $deg(\upsilon^{next}) > 1$ **and** $e^{ref} \neq \emptyset$ **do**
            $CE \leftarrow$ { edges connected to υ^{next} } $\backslash e^{ref}$
            $e^{next} \leftarrow \emptyset$
            forall $e^{ngb} \in CE$ **do**
                $\theta_{max} \leftarrow \theta_c; \theta \leftarrow \angle(e^{ref}, e^{ngb})$
                if $\theta > \theta_{max}$ **then**
                    $\theta_{max} \leftarrow \theta; e^{next} \leftarrow e^{ngb}$
                end
            end
            $e^{ref} \leftarrow e^{next}$
            if $e^{ref} \neq \emptyset$ **then**
                $\upsilon^{next} \leftarrow \upsilon_2$, where $e^{ref} = (\upsilon_2, \upsilon^{next})$
                $W \leftarrow W \cup \{e^{ref}\}$
            end
        end
    end
    $\Lambda \leftarrow \Lambda \cup \{W\}$ and $E \leftarrow E \setminus W$
end

Algorithm 1: Decomposing the Set of Edges of G_{Υ} Into Distinct Paths. A Vertex and an Edge Are Called Incident If the Vertex Is One of the Two Vertices the Edge Connects

FIGURE 4. (a) In the proximity of every junction-point, e.g. j_1 blue filled-circle, and on each sub-skeleton, e.g. ψ_1 green line, we define two decomposition intervals, $[t_s^+, t_e^+]$ and $[t_e^-, t_s^-]$, tracing ψ_1, from t_s^+ to t_e^+ and from t_s^- to t_e^- (red filled-circles). The lower and upper bounds of the intervals are two factors of the radius of the maximal inscribed ball at t_j, the green circle. (b) Decomposition intervals in the proximity of all junction-points j_1 and j_2 and for all sub-skeletons ψ_1, ψ_2, and ψ_3 are defined with the red filled-circles. Only in decomposition intervals, we are allowed to sweep the object. Arrows depict the sweeping direction to approach junction-points.

FIGURE 5. (a) Two nearly similar curves C_1 and C_2. Turquoise arrows represent $OM(C_1, C_2)$, and pink arrows represent $OM(C_2, C_1)$. (b) The average curve μ obtained from the orthogonal correspondence between C_1, an already visited curve, and C_2 a new cross-sectional curve.

such a point on the sub-skeleton that signed arc-length from t_j to t_s^+ (t_s^-) equals r_s $(-r_s)$. And t_e^+ (t_e^-) is such a point on the sub-skeleton that signed arc-length from t_j to t_e^+ (t_e^-) equals r_e $(-r_e)$. We have $t_s^+ < t_e^+ < t_j < t_e^- < t_s^-$.

The upper and lower thresholds may imply arc-distances outside parametrization limits of ψ. If the arch-length from $\psi(0)$ to $\psi(t_j)$ is smaller than r_s (r_e) we assign $t_s^+ = 0$ $(t_e^+ = 0)$. And if the arch-length from $\psi(t_j)$ to $\psi(1)$ is smaller than r_s (r_e) we assign $t_s^- = 1$ $(t_e^- = 1)$. Also, when a junction-point is at the either ends of a sub-skeleton, e.g., in a T-shape object, we define only one decomposition interval. Therefore, if $\psi(t_j) = 0$ $(\psi(t_j) = 1)$ the only interval that we define is $[t_e^-, t_s^-]$ $([t_s^+, t_e^+])$. Fig. 4b shows decomposition intervals in the proximity of j_1 and j_2 on sub-skeletons ψ_1, ψ_2, and ψ_3.

B. CRITICAL POINT
A critical point on a sub-skeleton is such a point that the cross-sectional contour of the object at this point changes substantially (Fig. 6). We use the Hausdorff metric to compare geometrical changes between cross-sectional contours
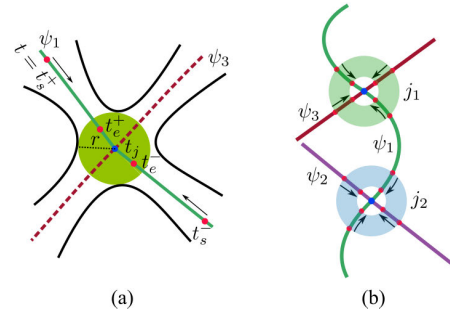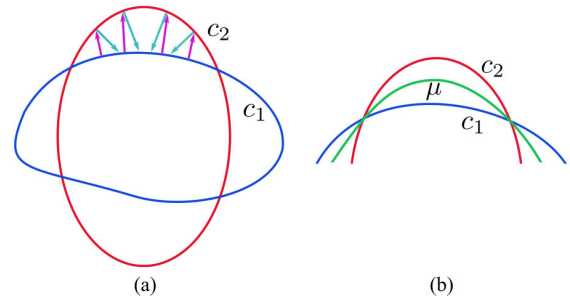
in a decomposition interval. The Hausdorff distance between two curves C_1 and C_2 is calculated as

$$\mathcal{H}(C_1, C_2) = \max\{\sup_{p \in C_1} \inf_{q \in C_2} d(p, q), \sup_{q \in C_2} \inf_{p \in C_1} d(p, q)\}, \tag{2}$$

where d(.) is the Euclidean distance between two points. We sweep $\partial\Omega$ by a cross-sectional plane $\mathcal{P} \subset \mathbb{R}^3$ to extract the cross-sectional contours. A cross-sectional plane $\mathcal{P}(t)$ is a plane orthogonal to ψ at every point t along ψ. The plane normal is equal to the tangent vector to ψ at point $\psi(t)$. We sweep $\partial\Omega$ by $\mathcal{P}$ along ψ in $[t_s^+, t_e^+]$ interval starting at t_s^+ toward t_e^+, and in $[t_e^-, t_s^-]$ interval starting at t_s^- toward t_e^-, as illustrated in Fig. 4. Let $\mathcal{P}(t)$ intersects $\partial\Omega$ at an inquiry point t. Since we assumed that $\partial\Omega$ is homeomorphic to a 2-sphere, the cross-sectional contour $C(\varsigma) : [0, 1] \rightarrow \mathbb{R}^2$ is a simple closed curve, where $C(0) = C(1)$. Translating $\mathcal{P}$ along $\psi(t)$ with t moving in decomposition intervals, we compare the Hausdorff distance between the cross-sectional contour at t denoted as C_t with the average of visited cross-sectional contours μ.
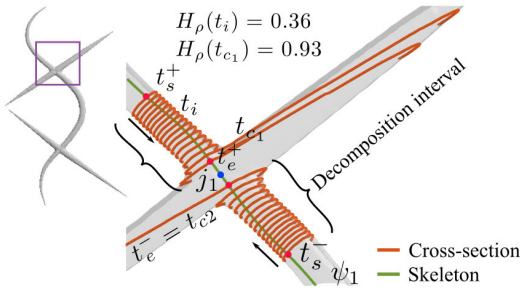
FIGURE 6. Sweeping the object surface along the sub-skeleton ψ_1 at junction-point j_1 (blue filled-circle) between $[t_s^+, t_e^+]$ and $[t_s^+, t_e^+]$ (red filled-circles). Sweeping directions are shown with arrows. At any decomposition interval, if $H_\rho < \theta_H$ the inquiry continues to the next point. If $H_\rho(t) \geq \theta_H$ the inquiry stops at t and the point is called a critical point. The critical point in the first interval is denoted as t_{c_1} and in the second interval is denoted as t_{c_2}.

To find the average curve μ between two nearly similar curves C_1 and C_2, we first need a one-to-one orthogonal mapping (OM) between C_1 and C_2. Consider that C_1 is parameterized by ς. To each point $C_1(\varsigma)$ of C_1, the $OM(C_1, C_2)$ associates the closest point $C_2(\varsigma)$ on C_2 that lies on the line passing through $C_1(\varsigma)$ and having for direction the normal $N(\varsigma)$ to C_1 at $C_1(\varsigma)$. Having this mapping, then each point $C_2(\varsigma)$ of C_2 may be expressed as the normal offset $C_1(\varsigma) + d(\varsigma)N(\varsigma)$ of $C_1(\varsigma)$. We say that $C_1(\varsigma)$ is the closest normal projection of $C_2(\varsigma)$ onto C_1 and can express C_2 as a deformation of C_1 completely defined by the normal displacement field $d(\varsigma)$ (see Fig. 5a) [36]. The average curve obtained by this orthogonal correspondence is asymmetric: $OM(C_1, C_2)$ is not necessarily equal to $OM(C_2, C_1)$. Therefore, we consistently take C_1 as an already visited curve, C_2 as the new cross-sectional curve, and define the average curve over $OM(C_1, C_2)$ (see Fig. 5).

We normalize the Hausdorff distance $\mathcal{H}(C_t, \mu)$ to the range $[0, 1]$ and denote it as $H_\rho(t)$. For that we first find a point interior to C_t denoted as κ. We define $\kappa \in \mathbb{R}^2$ to be the intersection of $\mathcal{P}$ and ψ at point t. Defining $d_{C_t}(\kappa) = \sup_{q \in C_t} d(\kappa, C_t)$, we write $H_\rho(t)$ as

$$H_\rho(t) = \frac{\mathcal{H}(C_t, \mu)}{\mathcal{H}(C_t, \mu) + d_{C_t}(\kappa)}. \quad (3)$$

We define a similarity threshold between cross-sectional contours as θ_H. While sweeping $\partial\Omega$ along ψ from t_s^+ (t_s^-) to t_e^+ (t_e^-), if $H_\rho(t) < \theta_H$, the inquiry continues to the next point. However, if $H_\rho(t) \geq \theta_H$ the inquiry stops at t and the point is called a critical point, denoted as t_{c_1} (t_{c_2}), as shown in Fig. 6. In $[t_s^+, t_e^-]$ ($[t_e^-, t_s^-]$), if at no inquiry point $H_\rho(t)$ exceeds θ_H, we define the t_{c_1} (t_{c_2}) as the point with minimum arc-distance r ($-r$) to $\psi(t_j)$ at which H_ρ is maximum.

VI. OBJECT RECONSTRUCTION

We cut the object at all critical points and decompose $\partial\Omega$ into n parts, n is the number of skeleton branches, and δ intersections, δ is the number of junction-points. We distinguish between an object part and an intersection such that the
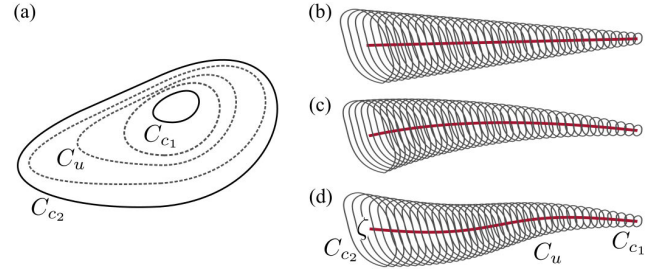

FIGURE 7. (a) Homotopy between two curves $C_{c_1}(\varsigma)$ and $C_{c_2}(\varsigma)$. Generalized cylinder along (b) a linear, (c) spline, and (d) sine interpolation between $\psi(t_{c_1})$ and $\psi(t_{c_2})$.

interior of an intersection includes a junction-point. The final decomposition step is to discard intersections and assign the same label to those object parts that are along the same sub-skeleton to obtain m semantic tubular components, m is the number of sub-skeletons. As we discard the intersections, we reconstruct the semantic tubular components using generalized cylinders. A generalized cylinder $\Phi(u, \varsigma)$: $[0, 1]^2 \to \mathbb{R}^3$ represents an elongated surface on an arbitrary axis and smoothly varying cross-sections [37]. In Cartesian coordinates x_1, x_2, x_3, the axis is parametrized by u as $\zeta(u) = (x_1(u), x_2(u), x_3(u))$ and cross-section boundary is represented as $C_u(\varsigma) = (x_1(u, \varsigma), x_2(u, \varsigma))$. To construct Φ, we apply a translational sweep along $\zeta(u)$ using closed simple curves $C_u(\varsigma)$ written as

$$\Phi(u, \varsigma) := \{\zeta(u) \in \mathbb{R}^3, C_u(\varsigma) \in \mathbb{R}^2 : u, \varsigma \in [0, 1]\}. \quad (4)$$

To obtain a parametric representation of generalized cylinders, it is convenient to employ a local coordinate system defined with the origin at each point of $\zeta(u)$. A convenient choice is the Frenet-Serret frame which is suitable for describing the kinematic properties of a particle moving along a continuous, differentiable curve in $\mathbb{R}^3$. The Frenet-Serret frame is an orthonormal basis composed of three unit vectors e_T, e_N, and e_B, where e_T is the unit tangent vector, and e_N and e_B are the unit normal and unit binormal vectors, respectively. By defining the cross-section in the Frenet-Serret frame, we form a parametric representation of generalized cylinders [38] as follows:

$$\Phi(u, \varsigma) = \zeta(u) + x_1(u, \varsigma)e_N(u) + x_2(u, \varsigma)e_B(u) \quad (5)$$

To define $C_u(\varsigma)$, we use a homotopy between two curves $C_{c_1}(\varsigma)$ and $C_{c_2}(\varsigma)$, where the curves are obtained by cross-sectioning the object surface at critical points t_{c_1} and t_{c_2}, respectively (see Fig. 7a). Let the simple closed curves $C_{c_1}(\varsigma)$ and $C_{c_2}(\varsigma)$ in $\mathbb{R}^2$ be homotopic with a continuous map h : $[0, 1]^2 \to \mathbb{R}^2$. So, we write:

$$h(0, \varsigma) = C_{c_1}(\varsigma), \quad h(1, \varsigma) = C_{c_2}(\varsigma), \quad \forall \varsigma \in [0, 1], \quad (6)$$

$$h(u, 0) = h(u, 1), \quad \forall u \in [0, 1], \quad (7)$$

where h is called a homotopy from $C_{c_1}(\varsigma)$ to $C_{c_2}(\varsigma)$. We denote a cross-section at a point along $\zeta(u)$ as $C_u := h(u, .)$. Note that, $\mathbb{R}^2$ is simply connected space. We use a linear

homotopy between $C_{c_1}(\varsigma)$ to $C_{c_2}(\varsigma)$ defined as:

$$h(u, \varsigma) = (1 - u)\, C_{c_1}(\varsigma) + u\, C_{c_2}(\varsigma), \tag{8}$$

where the computation on the right side is in $\mathbb{R}^2$. Equation (8) essentially indicates that we are moving from $C_{c_1}(\varsigma)$ to $C_{c_2}(\varsigma)$ along a straight line. To define the curve $\zeta(u)$, we use an interpolation between $\psi(t_{c_1})$ and $\psi(t_{c_2})$. Figs. 7b-d show Φ on different choices of ζ.

VII. EXPERIMENTAL RESULTS

In this section, we evaluate the effect of CSD parameters on object decomposition, present the applications and advantages of the proposed method in decomposing tubular objects, and show the performance of CSD applied on more general objects. We use the marching cubes algorithm [39] to compute a triangulated mesh of the object surface, visualizing voxel-based objects.

A. PARAMETER SETTING

In all our experiments, we fix the value of $\alpha_e = 1$ (section V-A), meaning that the distance of t_e^+ (t_e^-) to a junction-point is equal to the radius of the maximal inscribed ball at that junction-point. Here, we examine the effect of α_s, which determines t_s^+ (t_s^-) of the decomposition intervals (section V-A) for values equal to 10, 20, and 30. For that, we set the similarity threshold θ_H (section V-B) equal to 0.7. We use a linear interpolation to define ζ, the curve on which a generalized cylinder is defined (section VI), and set the value of the angular threshold θ_c (section IV-B) equal to 0°. Figs. 8a-c show the decomposition of the synthetic tubular object for α_s equals 10, 20, and 30, respectively. The decomposition/reconstruction at $\alpha_s = 10$ is more faithful to the original object as the critical points are detected close to the junctions. Increasing the value of α_s enlarges the decomposition interval. Therefore, at $\alpha_s = 30$, CSD detects the critical points distant from the junctions, resulting in a bigger reconstruction error compared to small values of α_s. It is worth noting that although setting α_s to small values provides more accurate decomposition/reconstruction results, it may also result in defining a critical point within an intersection. This can be the case when applying CSD to objects degraded with surface noise. The curve skeleton of an object with surface noise may not exactly lie in the center of the object, which means that the junction-point can be dislocated and the radius of the maximum inscribed ball at that junction-point be measured smaller than its true value. For α_s and α_e, we suggest values in range [3, 20] and [0.5, 2], respectively.

We also examine the effect of θ_H value, which is the similarity threshold between cross-sectional contours and μ. Figs. 9a-c show the decomposition of the tubular synthetic object at θ_H equals 0.6, 0.7, and 0.8, respectively. To better demonstrate the effect of θ_H, we set $\alpha_s = 30$, which we earlier showed this could result in a substantial decomposition error. We use a linear interpolation to define ζ and set $\theta_c = 0°$. At $\theta_H = 0.6$, CSD is sensitive to cross-sectional changes

(a) $\alpha_s = 10, \alpha_e = 1,$ (b) $\alpha_s = 20, \alpha_e = 1,$ (c) $\alpha_s = 30, \alpha_e = 1,$
    $\theta_H = 0.7$ $\theta_H = 0.7$ $\theta_H = 0.7$
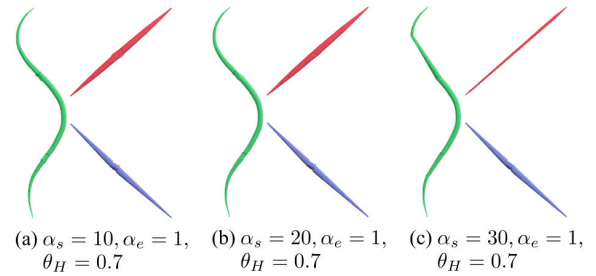
FIGURE 8. Decomposition of the synthetic tubular object at $\alpha_s = 10, 20, 30$ for fixed values of $\alpha_e = 1$ and $\theta_H = 0.7$. We use a linear interpolation to define ζ and set $\theta_c = 0°$. At $\alpha_s = 10$, the decomposition/reconstruction is in agreement with the original object because critical points are detected close to the junctions. Increasing the value of α_s enlarges the decomposition intervals, which may result in inaccuracy while decomposition/reconstruction, e.g., at $\alpha_s = 30$.

(a) $\alpha_s = 30, \alpha_e = 1,$ (b) $\alpha_s = 30, \alpha_e = 1,$ (c) $\alpha_s = 30, \alpha_e = 1,$
    $\theta_H = 0.6$ $\theta_H = 0.7$ $\theta_H = 0.8$
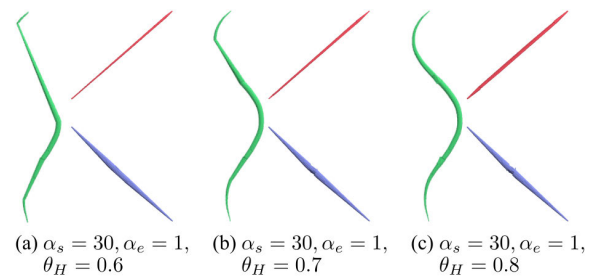
FIGURE 9. Decomposition of the synthetic tubular object at $\theta_H = 0.6, 0.7, 0.8$ for fixed values of $\alpha_s = 30$ and $\alpha_e = 1$. We use a linear interpolation to define ζ and set $\theta_c = 0°$. Increasing the θ_H value increases the CSD tolerance in dealing with gradient cross-sectional changes of the tubes. At $\theta_H = 0.8$, CSD recognizes the critical points near to junction-points, despite distant starting points from the junctions.

and does not tolerate the gradual increase of the tube diameter; hence critical points are detected distant from junction-points, and the reconstruction shows a low agreement with the original object. Increasing the value of θ_H to 0.7 increases the tolerance of CSD to cross-sectional changes. Therefore, despite distant starting points from junction-points, the reconstruction shows a better agreement to the original object, and at $\theta_H = 0.8$, the reconstructed object is faithful to the original object. Note that increasing θ_H elevates the tolerance of CSD to the cross-sectional changes quickly, e.g., at $\theta_H = 0.9$, the algorithm tolerates a nine times difference between a cross-section and μ, and at $\theta_H = 0.95$, it tolerates a 19 times difference. We suggest θ_H to be in the range [0.7, 0.85].

In the skeleton partitioning section (section IV-B), we showed that we merge two successive edges when the angle between them is bigger than θ_c. Therefore, by setting θ_c to big values, we emphasize the straightness of a path, but then the path may not be maximal-length. Fig. 10 shows how θ_c affects the number of semantic components. At $\theta_c = 0°$, all successive edges are allowed to merge even with acute angles; therefore we obtain a minimum number of object partitions with maximal-length paths (Fig. 10b). By increasing θ_c, only successive edges with a close-to-straight angle are allowed to be merged, which reduces the number of merges and increases the number of object partitions. Fig. 10c shows that
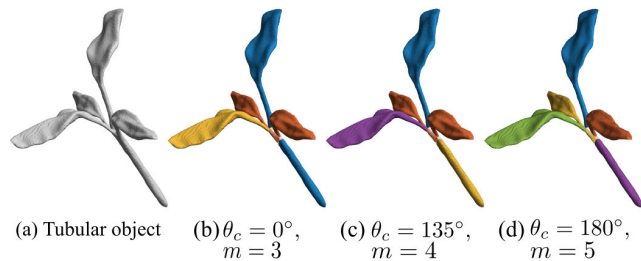
(a) Tubular object   (b) $\theta_c = 0°$, $m = 3$ (c) $\theta_c = 135°$, $m = 4$ (d) $\theta_c = 180°$, $m = 5$

FIGURE 10. The angle between two successive edges in a path should be bigger than θ_c to be merged. (a) A synthetic tubular object, size: 128 × 128 × 128 voxels. (b) Setting $\theta_c = 0°$ produces maximal-length paths; the minimum number of object parts $m = 3$. (c) At $\theta_c = 135°$, the number of object parts increases to $m = 4$. (d) At $\theta_c = 180°$, every edge in the skeleton graph is a path, hence producing the maximum number of semantic components, which is equal to the number of skeleton branches, $m = 5$.
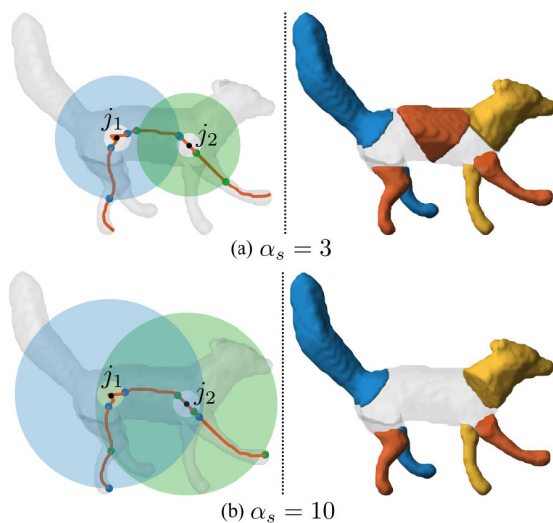

(a) $\alpha_s = 3$

(b) $\alpha_s = 10$

FIGURE 11. Overlapping of decomposition intervals when junction-points are adjacent. (a) Two junction-points (left panel) and two intersections of parts (right panel, grey sections) when α_s is equal to 3. (b) Two junction-points (left panel) and one intersection of parts (right panel, grey section) when α_s is equal to 10.

decomposition for $\theta_c = 135°$ yields four semantic components. At $\theta_c = 180°$, no successive edges are merged, and every edge in the skeleton graph corresponds with an object part. Setting $\theta_c > 180°$ generates the maximum number of object parts, equal to the number of skeleton branches (Fig. 10d).

We design CSD to have the same number of part intersections as the number of junction-points. However, when two junction-points appear adjacent on a sub-skeleton, we can merge them into one, depending on the value of α_s. Fig. 11a shows a four-leg with two intersections. For $\alpha_s = 3$, the decomposition interval around j_1 does not cover j_2, and the decomposition interval around j_2 does not cover j_1; therefore, the back of the four-leg, within its body, is decomposed. Fig. 11b shows that for $\alpha_s = 10$, the decomposition interval around j_1 includes j_2, and the decomposition interval around j_2 includes j_1, resulting in one object part intersection, where the four-leg body is the intersection of object parts.

B. AXON SEGMENTATION IN ELECTRON MICROSCOPY VOLUMES

The primary purpose of developing CSD is to segment tens of thousands of myelinated axons in electron microscopy volumes of white matter, whose sizes are approximately 4000 × 2000 × 1300 voxels. We generate a probability map of myelinated axons using deep convolutional neural networks (for details, we refer to [40]). We threshold the probability map, and using connected component analysis, we obtain a preliminary foreground segmentation of myelinated axons. Fig. 12a shows examples of myelinated axons after connected component analysis with an under-segmentation error(s): an axon intersects other axons or merges with the extra-axonal space. We apply CSD to evaluate every preliminary segmentation of myelinated axons for the under-segmentation error. If CSD recognizes an under-segmentation error, it decomposes the segmented component into its semantic parts. Fig. 12 shows the proposed decomposition of myelinated axons compared to the ACD (developed for point clouds) and skeleton-to-surface mapping approaches. To apply ACD on large objects, we first down-sample the point cloud representation of objects to 50 000 points, enabling the decomposition to be performed in a reasonable time (less than 10 minutes per object). Fig. 12b shows that ACD over-segments myelinated axons. We perform skeleton-to-surface mapping decomposition based on the Voronoi partitioning of surfaces, using Euclidean distance to skeleton branches (Fig. 12c). Because a curve skeleton captures the object geometry, skeleton-to-surface mapping decomposes an object close-to-semantic, but it does not recognize intersections of object parts and the boundary cuts are not correct. Fig. 12d shows our decomposition of myelinated axons, where CSD generates the correct number of semantic parts for under-segmented myelinated axons and reconstructs axons at intersections using generalized cylinders. Fig. 13 shows the complete segmentation of myelinated axons in a large electron microscopy volume, where CSD scans, decomposes, and reconstructs about 30 000 myelinated axons.

C. DECOMPOSITION OF VASCULAR NETWORKS

We compare our method to ACD and skeleton-to-surface mapping for the decomposition of a vascular network. Fig. 14a shows that ACD over-segments the vascular network. Fig. 14b shows that skeleton-to-surface mapping decomposes the object into 20 semantic components based on the Euclidean distance to skeleton benches, but the method does not identify intersections, yet the boundary cuts are not correct. For example, Fig. 14b (magnified box) shows that where the thin vessel (green partition) bends on the thick vessel (red partition), skeleton-to-surface mapping erroneously assigns a part of the thick vessel to the thin vessel, the part which is closer to the skeleton of the thin vessel. CSD decomposes the object into eight semantic components and reconstructs the object at intersections (Fig. 14c).
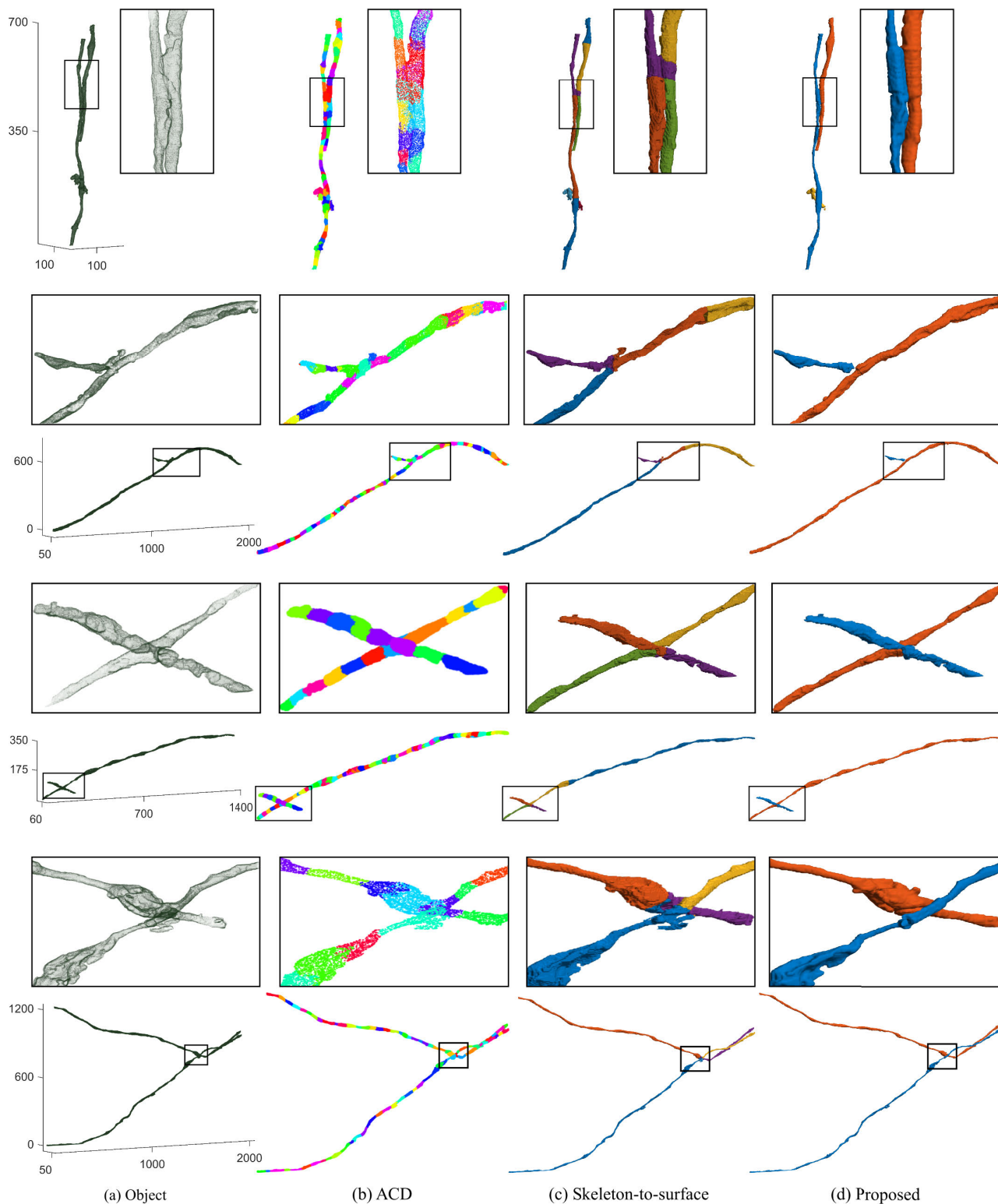
FIGURE 12. (a) Examples of foreground segmentation of myelinated axons with under-segmentation. (b) Decomposition using ACD [1]. The point cloud representation of objects is first down-sampled to 50 000 points to enable the decomposition task in a reasonable time; this method over-segments the objects. (c) Skeleton-to-surface mapping [21] based on Voronoi partitioning of the surface using skeleton branches. (d) CSD decomposition provides the correct number of semantic components in under-segmented myelinated axons. The objects are reconstructed at intersections using generalized cylinders. Objects inside boxes are magnified.
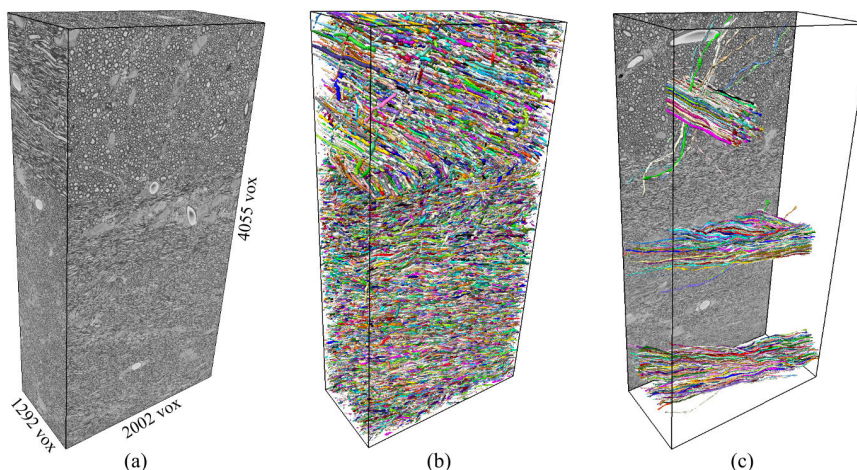
FIGURE 13. (a) A large electron microscopy volume of the white matter. The size of the volume is 4 055 × 2 002 × 1 292 voxels in x, y, and z directions, respectively. (b) A 3D rendering of myelinated axons (at one-third of the original resolution). CSD evaluates a preliminary segment for under-segmentation error(s), and if required, decomposes and reconstructs an under-segmented myelinated axon. (c) A 3D rendering of myelinated axons sampled at different locations illustrating the diversity of thickness and orientation in segmented axons.
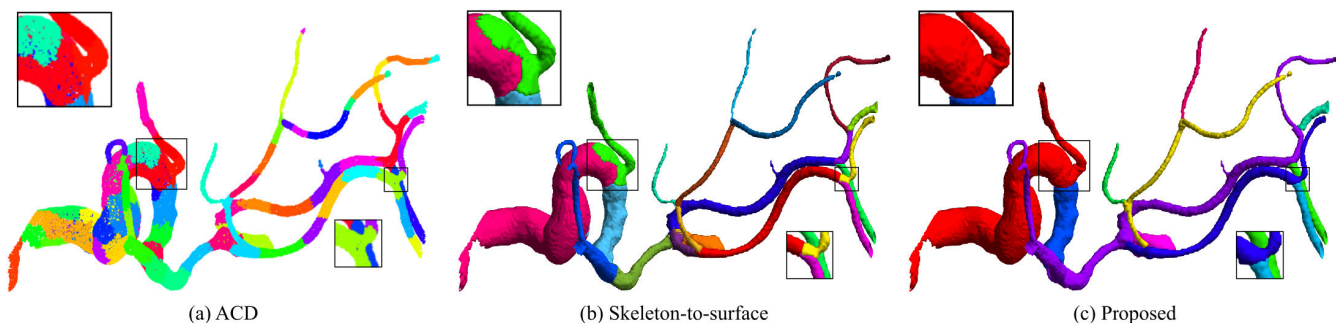

(a) ACD   (b) Skeleton-to-surface   (c) Proposed

FIGURE 14. (a) ACD [1] over-segments the vascular network. (b) Skeleton-to-surface mapping [21] decomposes the object into 20 semantic components, and the boundaries between these components are not accurate. (c) CSD decomposes the object into eight semantic components and reconstructs the object at intersections. Objects inside boxes are magnified. The 3D image of the vascular network is acquired from Colin Macdonald's GitHub page.

FIGURE 15. A gallery of CSD decomposition of synthetic objects.

D. DECOMPOSITION OF SYNTHETIC OBJECTS

To demonstrate the general applicability of the CSD algorithm, we examine the proposed CSD method on synthetic voxelized objects. The synthetic objects are from the Princeton segmentation benchmark database [41]. We voxelize meshes from the Princeton database using a ray intersection method described in [42]. The resolution of a voxelized object is determined using the bounding box of its OFF model; the bounding box values are normalized to range in (0, 1] then multiplied by 128. The resolution at each dimension is proportional to the length of the bounding box at that dimension, e.g., the dimension with the maximum length is represented by 128 voxels. Fig. 15 shows a gallery of decomposition on a mixture of objects with articulating parts, such

FIGURE 16. Human shape decomposition compared to the CSD decomposition results. The human decomposition of meshes into functional parts are treated as probabilistic ground truth; darker lines show places where more people placed a segmentation boundary.

as humans, octopuses, or pliers, and objects with moderate or small articulation, such as birds or fishes. Objects such as tables or airplanes include flat parts, which cannot be considered tubular. In Fig. 16, we qualitatively compare several of the CSD results to how humans decompose an object into functional parts (the darker the seam, the more people have chosen a cut along that edge [41]). For quantitative analysis, we compare ACD, GCD, and the proposed CSD method to the human decomposition over objects acquired from the Princeton database. We select two objects per category from the Princeton database, excluding categories that have objects with the genus bigger than zero, such as cups or vases, and categories that have an ambiguous skeleton, such as busts or mechs. The objects and their corresponding human segmentations are converted from mesh to voxel-based representation. To aggregate evaluation metrics over multiple human segmentations for the same model and multiple models for the same object category, we report averages per category (averages are computed first within each model, then the results are averaged within each object category). We report Rand error [43] and boundary error [44] and propose using the variance of information (VOI), which has a better discriminative error range than Rand error [45]. Table 1 shows that the proposed CSD algorithm outperforms both ACD and GCD methods. We stress that these results refer to the decomposition of the voxel-based objects (as obtained in biomedical imaging experiments), and we make no claims about the superiority of CSD when, for example, mesh-based representation of the surfaces would be the natural one.

E. DECOMPOSITION OF NOISY SYNTHETIC OBJECTS

We develop CSD to decompose voxel-based objects, e.g., objects extracted from biomedical images, where noise can degrade the object surface. To examine how noise affects decomposition techniques, as shown in Fig. 17a, we add impulse noise to the surface of an object for different noise density D_n values; $D_n = 0, 0.1, 0.35,$ and 0.6. Fig. 17b shows that ACD over decomposes the noise-free object to 64 parts and 82 parts when noise density equals 0.6. The GC decomposition of the noise-free object is approximately correct, but over-decomposes the noisy object at $D_n = 0.1$

TABLE 1. Comparison of decomposition techniques using Rand error (RE), the variance of information (VOI), and boundary error (BE) to human shape decomposition; smaller values are better. The average decomposition time (Time) of all objects reported in this table is presented as mean ± standard deviation. These decomposition techniques are implemented using different programming languages (ACD [1]: C++ and Matlab, GCD [2]: C++, and CSD: Python) and take different object representations as input (ACD: point cloud, GCD: mesh, and CSD: voxel).

Input model	Evaluation	ACD [1]	GCD [2]	Proposed
Human	RE	0.2421	0.2897	**0.2029**
	VOI	3.0345	**1.7049**	1.7360
	BE	0.4002	0.4145	**0.1703**
Glasses	RE	0.2402	**0.1650**	0.2394
	VOI	1.7264	**0.7111**	0.8842
	BE	0.0654	0.0506	**0.0088**
Airplane	RE	0.2588	**0.1450**	0.1868
	VOI	1.9981	**0.9944**	1.4765
	BE	0.1066	0.0678	**0.0677**
Ant	RE	0.2284	**0.0699**	0.0842
	VOI	2.4227	1.0315	**0.5978**
	BE	0.4930	0.1573	**0.0281**
Octopus	RE	0.3135	0.1657	**0.0215**
	VOI	3.0222	0.9684	**0.2418**
	BE	0.5433	0.1736	**0.0223**
Table	RE	0.0353	0.0444	**0.0109**
	VOI	0.4656	0.2476	**0.1082**
	BE	0.1530	0.0305	**0.0051**
Teddy	RE	0.3721	0.3530	**0.0621**
	VOI	3.2006	1.1708	**0.5542**
	BE	1.5860	0.2210	**0.1851**
Hand	RE	0.3625	0.3400	**0.2201**
	VOI	2.7837	**1.6841**	1.1728
	BE	0.5442	0.2967	**0.1739**
Pliers	RE	0.2399	0.0830	**0.0656**
	VOI	2.7210	0.6070	**0.5419**
	BE	0.2736	0.0936	**0.0386**
Fish	RE	0.5879	0.5140	**0.2232**
	VOI	2.6352	1.7495	**0.8839**
	BE	1.1104	0.4107	**0.1337**
Bird	RE	0.1764	0.2309	0.2208
	VOI	1.6971	**1.2629**	1.3959
	BE	1.0669	0.1431	**0.0901**
Armadillo	RE	0.2312	0.1897	**0.1870**
	VOI	2.9909	**1.3408**	1.5887
	BE	0.8717	0.3583	**0.3407**
Four-leg	RE	0.4082	0.3667	**0.1868**
	VOI	2.9593	1.9207	**1.0821**
	BE	0.6796	0.3486	**0.1738**
	Time	1287 ± 820	336 ± 98	**186 ± 42**

(Fig. 17c). GCD does not decompose the object at stronger noise levels when D_n equals 0.3 or 0.6. The proposed CSD method decomposes the objects at different noise levels with excellent performance, as shown in Fig. 17f. For the quantitative analysis, we compare decompositions against human segmentation using Rand error, VOI, and boundary error, as in Table 2. The ACD and GCD decomposition errors are high on different metrics while constantly low for the proposed CSD method.
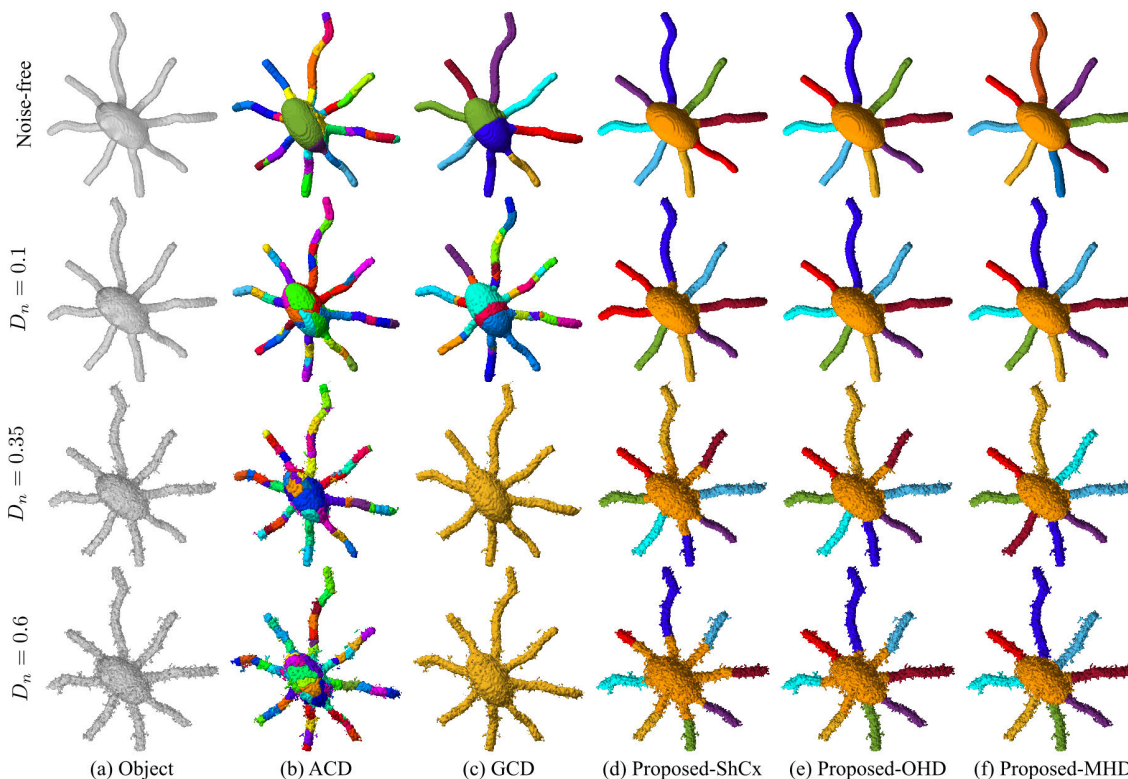
FIGURE 17. Decomposition of (a) ACD [1], (b) GCD [2], and the proposed CSD method (d-f) when the object surface is degraded with the impulse noise for the noise density D_n equals 0, 0.1, 0.35, and 0.6. The original Hausdorff distance as the similarity measure in the proposed CSD method is substituted with alternative metrics: (d) the shape context (ShCx) method [46], (e) original Hausdorff distance (OHD), and (f) modified Hausdorff distance (MHD) [47].

TABLE 2. Comparison of different techniques over the voxel-based object shown in Fig. 17 using Rand error (RE), the variance of information (VOI), and boundary error (BE). Smaller values indicate a closer decomposition to how humans decompose an object into functional parts. The average decomposition time (Time) of all objects reported in this table is presented as mean ± standard deviation. The GCD [2] method does not decompose objects at strong noise levels when D_n equals 0.3 or 0.6 and returns the object itself; therefore, we reported the GCD decomposition time for all experiments separately, showing the decomposition time of a failure case with ∞. These decomposition techniques are implemented using different programming languages (ACD [1]: C++ and Matlab, GCD: C++, and CSD: Python) and take different object representations as input (ACD: point cloud, GCD: mesh, and CSD: voxel).

Noise level	Evaluation	ACD [1]	GCD [2]	Proposed-ShCx	Proposed-OHD	Proposed-MHD
	RE	0.3633	0.2659	0.0276	**0.0211**	0.0263
Noise free	VOI	2.9877	1.2150	0.3046	**0.2482**	0.2940
	BE	0.5377	0.1976	0.0213	**0.0157**	0.0204
	RE	0.4159	0.3021	0.0530	**0.0207**	0.0249
$D_n = 0.1$	VOI	3.5099	1.9760	0.5486	**0.2424**	0.2819
	BE	0.7263	0.4015	0.0795	**0.0681**	0.0705
	RE	0.4677	0.4951	0.1064	0.0432	**0.0360**
$D_n = 0.35$	VOI	3.7282	1.7636	0.7965	0.3702	**0.3688**
	BE	1.1797	0.7403	0.2295	**0.2248**	0.2304
	RE	0.4221	0.5567	0.2036	0.0999	**0.0250**
$D_n = 0.6$	VOI	3.9624	1.9533	1.2451	0.7191	**0.2787**
	BE	0.9429	0.4605	0.3615	0.3581	**0.3569**
	Time	1302 ± 438	[360, 482, ∞, ∞]	1021 ± 468	**191 ± 26**	193 ± 25

F. CROSS-SECTIONAL SIMILARITY METRIC

To define a critical point in section V-B, we use the Hausdorff distance as defined in (2) to compare geometrical changes between cross-sectional contours. The Hausdorff distance can be sensitive to surface noise, showing a mismatch between cross-sectional contours that belong to the

same object part. Figs. 17d-f show the CSD decomposition performance, substituting the original Hausdorff distance with alternative shape matching techniques: the modified Hausdorff distance [47] and shape context metric [46]. We set θ_H equal to 0.8 for the original and modified Hausdorff distances and set a similarity threshold of 0.5 for the
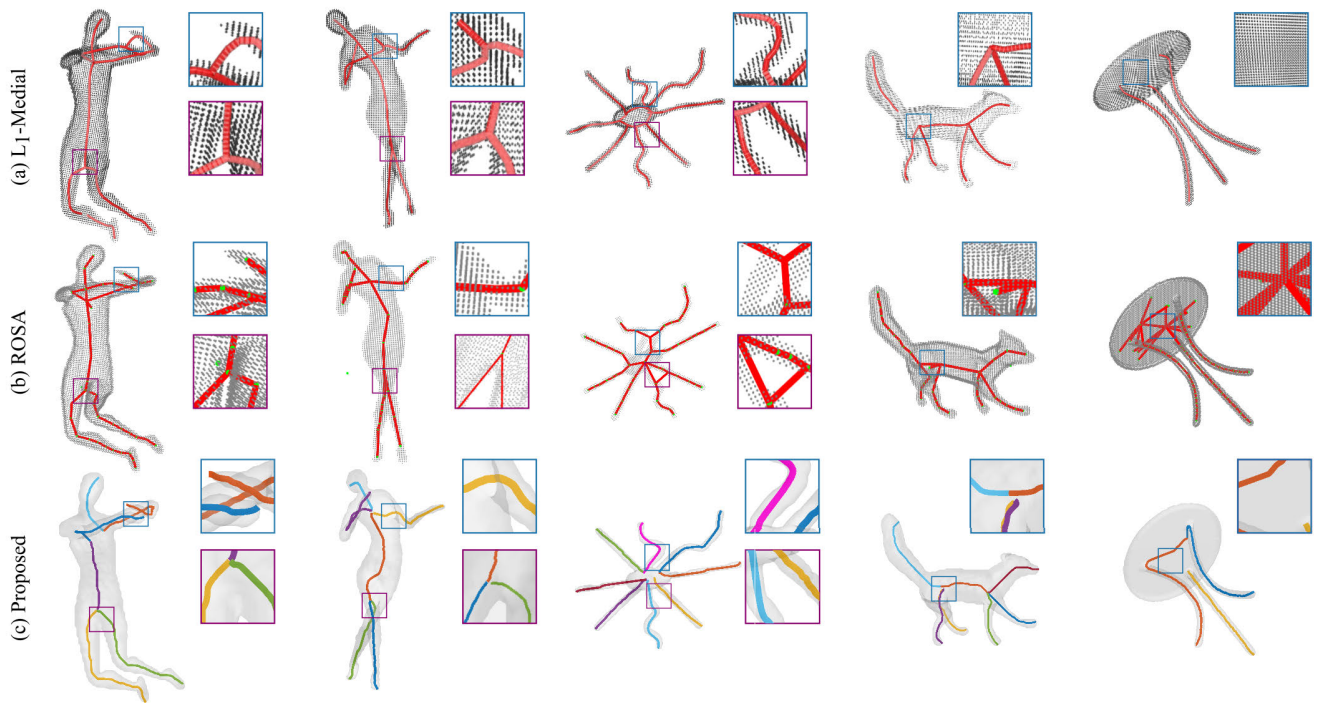
FIGURE 18. Skeletonization using (a) L1-medial [30], (b) ROSA [31], and (c) CSD distance-based techniques. L1-medial and ROSA may incorrectly form cycles in the skeleton of genus zero objects when two object parts are close, while the CSD skeletonization generates separate branches for different object parts. Also, L1-medial and ROSA do not necessarily stay within the object, while the CSD distance-based skeletonization approach penalizes the skeleton where the path moves away from the center of the object. Objects inside boxes are magnified.

shape context metric. The surface of the synthetic object in Fig. 17a is degraded with the impulse noise for the noise density D_n equals 0, 0.1, 0.35, and 0.6. In Table 2, we quantitatively evaluate decompositions using different cross-sectional similarity metrics and at different noise levels to the human segmentation using Rand error, VOI, and boundary error. Results demonstrate that different CSD similarity metrics yield excellent decompositions for the noise-free object, where the original Hausdorff distance performs better than the shape context metric and modified Hausdorff distance. With increasing the noise density, however, modified Hausdorff distance performs better than the shape context and the original Hausdorff metrics. The modified Hausdorff distance produces an excellent decomposition across noise densities.

G. CHOICE OF SKELETONIZATION TECHNIQUE

The skeleton partitioning step guides the CSD algorithm for semantic decomposition; therefore, the quality of the skeletonization itself is crucial for the decomposition. We compare the CSD skeletonization approach to the L1-medial skeletonization and ROSA techniques in terms of the topological correctness and centeredness of the extracted skeletons. We compare these skeletonization techniques on synthetic voxel-based objects from McGill 3D Shape Benchmark [6]. Fig. 18 shows that when two object parts appear very close, L1-medial (Fig. 18a human hand) and ROSA (Fig. 18b octopus) merge the object parts and form a cycle in genus zero objects. The CSD skeletonization, see Fig. 18c, generates

separate branches for different object parts. Also, flat object parts, such as tabletop, do not possess a reasonably meaningful curve skeleton, but CSD skeletonization generates a more meaningful skeleton than the other two methods. In terms of centeredness, L1-medial and ROSA do not stay within the object. The CSD distance-based skeletonization approach penalizes the skeleton where the path moves away from the center of the object. Table 3 shows the computation time spent on skeletonization with L1-medial, ROSA, and CSD methods. The CSD skeletonization is faster than the other two methods, rendering it suitable for large objects.

H. COMPUTATION TIME

The time complexity of the sub-voxel precise skeletonization is $O(n N_\Omega \log N_\Omega)$, where n is the number of skeleton branches, and N_Ω is the number of voxels in a discrete Ω. The $N_\Omega \log N_\Omega$ factor is from the fast marching algorithm [35]. The time complexity to determine a critical point is $O(N_p)$, where N_p is the number of inquiry points that we check for the cross-sectional changes in a decomposition interval. Defining the critical points is independent of N_Ω. The complexity of the method is measured through the number of basic arithmetic operations performed; other factors that may also influence the execution time, such as the number of memory accesses or memory consumption, have not been considered.

A fair comparison between the computation times of different decomposition techniques by the wall clock time requires the same constraints for all techniques. Providing

TABLE 3. The skeletonization time for objects in Fig. 18 using L1-medial [30], ROSA [31], and the proposed CSD method. These methods are implemented using different programming languages (L1-medial: C++, ROSA: C++ and Matlab, and CSD: Python). L1-medial and ROSA take point clouds as input, while CSD takes voxel-based objects as input. Therefore, the times reported are not directly comparable but provide insight into the speed of skeletonization algorithms, which is important for our segmentation application of large objects. For L1-medial and ROSA, we report the number of points (pts). For CSD, we report the number of voxels representing objects; the bounding box of objects is 128 × 128 × 128 voxels.

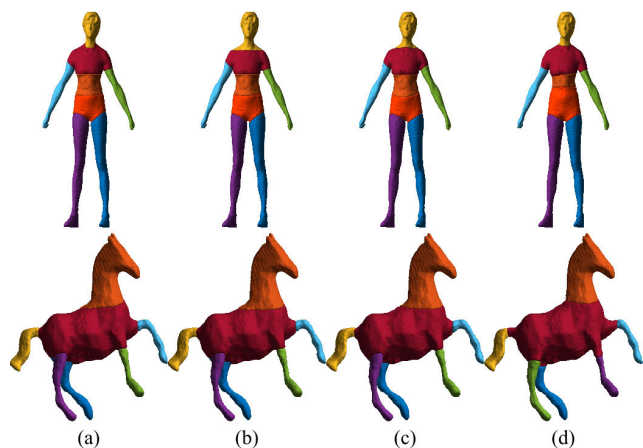| Model	Evaluation	L1-medial [30]	ROSA [31]	Proposed
Human	Size	1 000 pts	13 189 pts	30 785 vox
	Time (s)	37	549	**4**
Human	Size	1 000 pts	11 107 pts	24 897 vox
	Time (s)	31	403	**4**
Octopus	Size	1 000 pts	6 415 pts	9 179 vox
	Time (s)	19	144	**6**
Four-leg	Size	1 000 pts	11 878 pts	30 490 vox
	Time (s)	31	465	**4**
Table	Size	1 000 pts	19 499 pts	42 323 vox
	Time (s)	80	1 333	**4**

FIGURE 19. Sampling object sub-skeletons to reduce the decomposition time. The decomposition of a human and a four-leg at (a) $sf = 1$, (b) $sf = 4$, (c) $sf = 16$, and (d) $sf = 64$.

such constraints is challenging because different decomposition techniques use different programming languages or take different object representations as input. In spite of that, we demonstrate the average decomposition time of ACD, GCD, and the proposed CSD method in Tables 1 and 2. The average decomposition time of the proposed CSD is shorter than ACD and GCD in all our experiments: the average decomposition time of synthetic objects is 3 m for CSD, 21 m for ACD, and 5 m for GCD. Note that, although the computation times of the GCD and proposed CSD methods are close, we have not been able to decompose objects acquired from biomedical image datasets, e.g., axons (Fig. 12) and the vascular network (Fig. 14), using GCD within a day.

We also propose to reduce the CSD computation time by reducing the number of inquiry points N_p. For that, we propose to sub-sample the sub-skeletons by a sampling factor sf, as shown in Fig. 19. Increasing sf reduces the computation

TABLE 4. Evaluation of how sampling the sub-skeletons for sf equals 1, 4, 16, and 64 affects decomposition results of objects in Fig. 19 using Rand error (RE), the variance of information (VOI), and boundary error (BE), considering decomposition at $sf = 1$ as the ground truth. The decomposition results are achieved on a 4-core Intel CPU 3.41 GHz computer with 64 GB RAM using Python 3.6.

Model	Evaluation	$sf = 1$	$sf = 4$	$sf = 16$	$sf = 64$
Human	RE		**0.0140**	0.0203	0.0340
	VOI		**0.2507**	0.4082	0.5511
	BE		**0.0496**	0.0915	0.1797
	Time (s)	108.1	30.5	9.2	**4.5**
Four-leg	RE		**0.0319**	0.0359	0.0519
	VOI		**0.2149**	0.2775	0.3971
	BE		**0.0600**	0.0652	0.0714
	Time (s)	174.0	48.3	14.1	**4.9**

time linearly while affecting decomposition results minimally. Fig. 19 shows the decomposition of three objects at sf equals 1, 4, 16, and 64. To evaluate the effect of sampling the sub-skeletons on decomposition results, we compared decomposition results over Rand error, VOI, and boundary error, considering decomposition at $sf = 1$ as the ground truth. Table 4 shows that decomposition by a factor of four substantially reduces the computation time, whereas the evaluation metrics worsen minimally. Reducing the computation is important when dealing with big voxel-based objects. For example, on a $2 \times$ Intel Xeon E5 2 630 CPU 2.4 GHz machine with 512 GB RAM using Python 3.6, the skeletonization of the myelinated axon shown in the first row of Fig. 12 ($N_\Omega = 395\,594$) consumes 117 s and defining its critical points 353 s, and sampling the sub-skeletons by $sf = 5$ reduces the decomposition time to 75 s.

VIII. CONCLUSION
In this article, we proposed the application of 3D shape decomposition in image segmentation. We presented the novel CSD algorithm to decompose and reconstruct undersegmented tubular objects. The CSD method is guided by the curve skeleton decomposition, decomposing a tubular object into maximal-length, approximately straight parts. The object is cut at the intersection of parts using translational sweeps and reconstructed by generalized cylinders. We demonstrated the application of CSD on biomedical imaging volumes and synthetic objects. In particular, we applied CSD as instance segmentation to deep learning-based semantic segmentation of myelinated axons. Hundreds of thousands of myelinated axons were automatically evaluated for undersegmentation error, and under-segmented myelinated axons were decomposed into their constituent axons, using the same parameter values for all objects in all electron microscopy datasets. We showed that CSD outperforms state-of-the-art techniques in decomposing voxel-based objects and is robust to severe surface noise. CSD is highly parallelizable, substantially reducing the computation time of the segmentation in large biomedical imaging datasets. The proposed CSD algorithm allows for including the cylindricity as a global

shape-objective for a fast 3D segmentation of tubular objects in large biomedical imaging datasets.

COMPETING INTERESTS
The authors declare that they have no conflict of interest.

AVAILABILITY OF DATA AND MATERIAL
The source code of the CSD algorithm is available at https://github.com/aAbdz/CylShapeDecomposition.

REFERENCES
[1] O. V. Kaick, N. Fish, Y. Kleiman, S. Asafi, and D. Cohen-OR, "Shape segmentation by approximate convexity analysis," *ACM Trans. Graph.*, vol. 34, no. 1, pp. 1–11, Dec. 2014.

[2] Y. Zhou, K. Yin, H. Huang, H. Zhang, M. Gong, and D. Cohen-Or, "Generalized cylinder decomposition," *ACM Trans. Graph.*, vol. 34, no. 6, pp. 1–14, Nov. 2015.

[3] O. K.-C. Au, C.-L. Tai, H.-K. Chu, D. Cohen-Or, and T.-Y. Lee, "Skeleton extraction by mesh contraction," in *Proc. ACM SIGGRAPH Papers*, 2008, pp. 1–10.

[4] S. Berretti, A. Del Bimbo, and P. Pala, "3D mesh decomposition using reeb graphs," *Image Vis. Comput.*, vol. 27, no. 10, pp. 1540–1554, Sep. 2009, doi: 10.1016/j.imavis.2009.02.004.

[5] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhedral surface decomposition with applications," *Comput. Graph.*, vol. 26, no. 5, pp. 733–743, Oct. 2002.

[6] K. Siddiqi, J. Zhang, D. Macrini, A. Shokoufandeh, S. Bouix, and S. Dickinson, "Retrieving articulated 3-D models using medial surfaces," *Mach. Vis. Appl.*, vol. 19, no. 4, pp. 261–275, Jul. 2008.

[7] M. Goyal, S. Murugappan, C. Piya, W. Benjamin, Y. Fang, M. Liu, and K. Ramani, "Towards locally and globally shape-aware reverse 3D modeling," *Comput.-Aided Des.*, vol. 44, no. 6, pp. 537–553, Jun. 2012, doi: 10.1016/j.cad.2011.12.004.

[8] M. Averkiou, V. G. Kim, Y. Zheng, and N. J. Mitra, "ShapeSynth: Parameterizing model collections for coupled shape exploration and synthesis," *Comput. Graph. Forum*, vol. 33, no. 2, pp. 125–134, May 2014.

[9] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by example," in *Proc. SIGGRAPH*, 2004, pp. 652–663.

[10] G. D. Rubin, J. Leipsic, U. J. Schoepf, D. Fleischmann, and S. Napel, "CT Angiography after 20 Years: A transformation in cardiovascular disease characterization continues to advance," *Radiology*, vol. 271, no. 3, pp. 633–652, Jun. 2014.

[11] A. Abdollahzadeh, I. Belevich, E. Jokitalo, J. Tohka, and A. Sierra, "Automated 3D axonal morphometry of white matter," *Sci. Rep.*, vol. 9, no. 1, Dec. 2019, Art. no. 6084.

[12] Z. Zheng, J. Lauritzen, E. Perlman, and C. Robinson, "A complete electron microscopy volume of the brain of adult Drosophila melanogaster," *Cell*, vol. 174, no. 3, pp. 730–743, 2018, doi: 10.1016/j.cell.2018.06.019.

[13] A. Lucchi, K. Smith, R. Achanta, G. Knott, and P. Fua, "Supervoxel-based segmentation of mitochondria in EM image stacks with learned shape features," *IEEE Trans. Med. Imag.*, vol. 31, no. 2, pp. 474–486, Feb. 2012.

[14] J. Nunez-Iglesias, R. Kennedy, S. M. Plaza, A. Chakraborty, and W. T. Katz, "Graph-based active learning of agglomeration (GALA): A Python library to segment 2D and 3D neuroimages," *Frontiers Neuroinform.*, vol. 8, pp. 1–6, Apr. 2014.

[15] J. Funke, F. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga, "Large scale image segmentation with structured loss based deep learning for connectome reconstruction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1669–1680, Jul. 2019.

[16] P. D. Simari and K. Singh, "Extraction and Remeshing of Ellipsoidal Representations from Mesh Data," in *Proc. Graph. Interface Victoria, British Columbia, Canadian Hum.-Comput. Commun. Soc.*, 2005, pp. 161–168.

[17] R. Raab, C. Gotsman, and A. Sheffer, "Virtual woodwork: Making toys from geometric models," *Int. J. Shape Model.*, vol. 10, no. 1, pp. 1–29, Jun. 2004.

[18] M. Mortal, G. Patanè, M. Spagnuolo, B. Falcidienot, and J. Rossignac, "Plumber: A method for a multi-scale decomposition of 3D shapes into tubular primitives and bodies," in *Proc. ACM Symp. Solid Modeling. Appl.*, 2004, pp. 339–344.

[19] S. Asafi, A. Goren, and D. Cohen-Or, "Weak convex decomposition by lines-of-sight," *Eurograph. Symp. Geometry Process.*, vol. 32, no. 5, pp. 23–31, 2013.

[20] X. Li, T. W. Toon, and Z. Huang, "Decomposing polygon meshes for interactive applications," in *Proc. Symp. Interact. 3D Graph.*, New York, NY, USA, 2001, pp. 35–42.

[21] D. Reniers, J. van Wijk, and A. Telea, "Computing multiscale curve and surface skeletons of genus 0 shapes using a global importance measure," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 2, pp. 355–368, Mar. 2008.

[22] M. Livesu, M. Attene, G. Patané, and M. Spagnuolo, "Explicit cylindrical maps for general tubular shapes," *Computer-Aided Design*, vol. 90, pp. 27–36, Sep. 2017, doi: 10.1016/j.cad.2017.05.002.

[23] E. Kalogerakis, A. Hertzmann, and K. Singh, "Learning 3D mesh segmentation and labeling," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 1–12, Jul. 2010, doi: 10.1145/1778765.1778839.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. 30th IEEE Conf. Comput. Vis. Pattern Recognit.*, Jan. 2017, pp. 77–85.

[25] F. Yu, K. Liu, Y. Zhang, C. Zhu, and K. Xu, "PartNet: A recursive part decomposition network for fine-grained and hierarchical shape segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9483–9492.

[26] B. Deng, K. Genova, S. Yazdani, S. Bouaziz, G. Hinton, and A. Tagliasacchi, "CvxNet: Learnable convex decomposition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 31–41.

[27] I. Biederman, "Recognition-by-components: A theory of human image understanding an analogy between speech and object perception," *Psychol. Rev.*, vol. 5, no. 2, pp. 115–147, 1917.

[28] P. D. Grünwald, *The Minimum Description Length Principle* (Adaptive Computation and Machine Learning). Cambridge, MA, USA: MIT Press, 2007.

[29] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications, and algorithms," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 3, pp. 530–548, May 2007.

[30] H. Huang, S. Wu, D. Cohen-Or, M. Gong, H. Zhang, G. Li, and B. Chen, "L1-medial skeleton of point cloud," *ACM Trans. Graph.*, vol. 32, no. 4, p. 1, Jul. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2461912.2461913

[31] A. Tagliasacchi, H. Zhang, and D. Cohen-Or, "Curve skeleton extraction from incomplete point cloud," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–9, Jul. 2009.

[32] A. Lieutier, "Any open bounded subset of Rn has the same homotopy type as its medial axis," *CAD Comput. Aided Des.*, vol. 36, no. 11, pp. 1029–1046, 2004.

[33] M. Hassouna and A. Farag, "Robust centerline extraction framework using level sets," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Aug. 2005, pp. 458–465. [Online]. Available: http://ieeexplore.ieee.org/document/1467303/

[34] R. Van Uitert and I. Bitter, "Subvoxel precise skeletons of volumetric data based on fast marching methods," *Med. Phys.*, vol. 34, no. 2, pp. 627–638, Jan. 2007.

[35] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts.," *Proc. Nat. Acad. Sci. USA*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.

[36] F. Chazal, A. Lieutier, and J. Rossignac, "Projection-homeomorphic surfaces," in *Proc. ACM Symp. Solid Phys. Model. (SPM)*, 2005, pp. 9–14.

[37] U. Shani and D. H. Ballard, "Splines as embeddings for generalized cylinders," *Comput. Vis., Graph., Image Process.*, vol. 27, no. 2, pp. 129–156, Aug. 1984.

[38] D. H. Ballard and C. M. Brown, *Computer Vision*, 1st ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1982.

[39] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," in *Proc. 14th Annu. Conf. Comput. Graph. Interact. Techn.*, 1987, pp. 163–169. [Online]. Available: http://portal.acm.org/citation.cfm?doid=37401.37422

[40] A. Abdollahzadeh, I. Belevich, E. Jokitalo, A. Sierra, and J. Tohka, "DeepACSON: Automated Segmentation of White Matter in 3D Electron Microscopy," *bioRxiv*, vol. 2019, Jan. 2019, Art. no. 828541.

[41] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–12, Aug. 2009.

[42] S. Patil and B. Ravi, "Voxel-based representation, display and thickness analysis of intricate shapes," in *Proc. 9th Int. Conf. Comput. Aided Design Comput. Graph. (CAD-CG05)*, 2005, pp. 415–422. [Online]. Available: http://ieeexplore.ieee.org/document/1604669/

[43] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, p. 846, Dec. 1971. [Online]. Available: https://www.jstor.org/stable/2284239?origin=crossref

[44] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, Jul. 2001, pp. 416–423. [Online]. Available: http://ieeexplore.ieee.org/document/937655/

[45] J. Nunez-Iglesias, R. Kennedy, T. Parag, J. Shi, and D. B. Chklovskii, "Machine learning of hierarchical clustering to segment 2D and 3D images," *PLoS ONE*, vol. 8, no. 8, Aug. 2013, Art. no. e71715.

[46] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002. [Online]. Available: http://ieeexplore.ieee.org/document/993558/

[47] M.-P. Dubuisson and A. K. Jain, "A modified hausdorff distance for object matching," in *Proc. 12th Int. Conf. Pattern Recognit.*, 2002, pp. 566–568. [Online]. Available: http://ieeexplore.ieee.org/document/576361/

ALI ABDOLLAHZADEH received the master's degree in electrical engineering from the Tampere University of Technology, Finland, in 2016. He is currently pursuing the Ph.D. degree with the A. I. Virtanen Institute for Molecular Sciences, University of Eastern Finland, Kuopio, Finland. His research interests include shape analysis, image processing, machine learning, and their applications in large electron- and light-microscopy image volumes of the brain.

ALEJANDRA SIERRA received the Ph.D. degree in biochemistry from the Autonomous University of Madrid, Spain, in 2006. Since then, she has been working with the A. I. Virtanen Institute for Molecular Sciences, University of Eastern Finland, Kuopio, Finland, where she held both Academy of Finland Postdoctoral and Research Fellow positions and currently the Research Director. Her research interest includes the validation and development of magnetic resonance imaging techniques by incorporating microscopic tissue information in the healthy and diseased brain.

JUSSI TOHKA received the Ph.D. degree in signal processing from the Tampere University of Technology, Finland, in 2003. He was a Postdoctoral Fellow with the University of California at Los Angeles, Los Angeles, USA. He held an Academy Research Fellow position at the Department of Signal Processing, Tampere University of Technology. He was a CONEX Professor with the Department of Bioengineering and Aerospace Engineering, Universidad Carlos III de Madrid, Spain. He is currently with the A. I. Virtanen Institute for Molecular Sciences, University of Eastern Finland, Kuopio, Finland. His research interests include machine learning, image analysis, and pattern recognition and their applications to brain imaging.

● ● ●