

RESEARCH ARTICLE

Medial-ABC: an algorithm for the correspondence between myocardium and coronary artery mesh models based on the medial axis of coronary artery

Jehyun Cha¹, Joonghyun Ryu², Jin-Ho Choi^{3,4,*} and Deok-So Kim^{1,2,*}

¹School of Mechanical Engineering, Hanyang University, Seoul 04763, South Korea; ²Voronoi Diagram Research Center, Hanyang University, Seoul 04763, South Korea; ³School of Medicine, Sungkyunkwan University, Suwon, Gyeonggi-do 16419, South Korea and ⁴Department of Emergency Medicine, Samsung Medical Center, Seoul 06351, South Korea

*Co-corresponding authors: Deok-So Kim (dskim@hanyang.ac.kr) and Jin-Ho Choi (jhchoimd@gmail.com)

Abstract

The role of coronary arteries is to supply sufficient blood to myocardium. Obstruction of coronary arteries limits blood supply and causes myocardial ischemia or acute myocardial infarction, a major cause of human death. Hence, the quantification of the regional amount of heart muscle subtended by obstructed coronary arteries is of critical value in clinical medicine. However, conventional methods are inaccurate and frequently disagree with clinical practice. This study proposes a novel medial-axis-based correspondence (Medial-ABC) algorithm to find the correspondence between myocardium and coronary artery in order to segment regional myocardium at risk subtended by any potentially obstructed coronary artery. Given the triangular mesh models of coronary artery and myocardium, the proposed algorithm (i) computes the medial axis of coronary artery, (ii) finds the correspondence using the medial axis of coronary artery, and (iii) segments the coronary artery and myocardium. The proposed algorithm provides a robust mathematical linkage between myocardium at risk and supplying coronary arteries so that ischemic myocardial regions can be accurately identified. Hence, both the extent and severity of myocardial ischemia can be quantified effectively, efficiently, and accurately. Furthermore, the constructed mesh model of segmented coronary artery and myocardium can be post-processed for applications such as building optimization models of cardiac systems. The CardiacVis program, which implements the Medial-ABC algorithm, is freely available at Voronoi Diagram Research Center (<http://voronoi.hanyang.ac.kr/software/cardiacvis>) and will be an invaluable tool for quantitative patient-specific risk stratification in clinical practice.

Keywords: segmentation; cardiac computed tomography; Voronoi diagram; constrained Delaunay triangulation; computational geometry; obstruction of blood; myocardial ischemia and heart disease

1. Introduction

Coronary artery (CA) disease is the number one cause of death worldwide and entrusts a huge socioeconomic burden. Statistics reported the deaths of 7.4 million people from coronary

artery disease every year (World Health Organization, 2012). Atherosclerotic obstruction of CA blocks oxygenated blood supply to regional left ventricular myocardium (heart muscle), which causes severe myocardial ischemia or acute myocardial infarction, i.e. a heart attack. Note that left ventricle (LV) is

Received: 6 March 2020; Revised: 13 May 2020; Accepted: 20 June 2020

© The Author(s) 2020. Published by Oxford University Press on behalf of the Society for Computational Design and Engineering. This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits non-commercial re-use, distribution, and reproduction in any medium, provided the original work is properly cited. For commercial re-use, please contact journals.permissions@oup.com

the major muscle responsible for the most mechanical function of heart. Localizing and assessing the extent of regional myocardium at risk subtended by obstructed CA is of critical importance in the diagnosis and decision of treatment (Bruyne et al., 2014). A 17-piece LV model based on 2D images is currently used by clinical guidelines as recommended by the American Heart Association (Cerqueira et al., 2002).

However, the model does not reflect an individual structural variation of both CA and LV and frequently produces inaccurate assignments or disagreements between supplying CA and regional LV receiving blood (Ortiz-Pérez et al., 2008; Javadi et al., 2010). The establishment of an accurate and robust linkage between LV territory and supplying CA is required for the optimal diagnosis and treatment of CA disease (Bruyne et al., 2012; Kim et al., 2016, 2017; Bae et al., 2018; Han et al., 2018). The clinical outcome of patients with heart disease is mostly determined by the amount of functioning myocardium. Loss of myocardium results in decreased heart function, which is known as heart failure. Physicians can open/dilate the occluded/narrowed artery, which is known as percutaneous coronary intervention using stent implantation. Millions of these procedures are being performed annually worldwide.

There is an important and yet unsolved clinical issue in this procedure: How to determine the treatment target? Currently, most physicians determine the treatment target based on their experience and clinical expertise, which inevitably leads to non-negligible interoperator difference. Here, we attempt geometrical and mathematical interpretation of coronary artery disease.

In this study, we present a medial-axis-based correspondence (Medial-ABC) algorithm to find the correspondence between myocardium and coronary artery using the medial axis of coronary artery. Given the triangular mesh models of coronary artery and myocardium, the Medial-ABC algorithm first computes the medial axis of coronary artery and then finds the correspondence using the medial axis of coronary artery. Then, it is easy to segment a myocardial geometric model of triangular mesh so that ischemic LV region can be accurately identified for each individual patient. The geometric cardiac models are useful for the visualization (Cai, Ye, Chui, & Anderson, 2003; Meyer-Spradow, Stegger, Döring, Ropinski, & Hinrichs, 2008; Borkin et al., 2011; Glaber, Lawonn, Hoffmann, Skalej, & Preim, 2014), analysis (Oeltze, Doleisch, Hauser, Muigg, & Preim, 2007; Termeer et al., 2008), and simulation of diverse features of cardiac systems (Termeer, Bescós, and Breeuwer, 2007; Xiong et al., 2017). The Medial-ABC algorithm segments the mesh model of LV into subregions where each corresponds to a piece of CA and/or a concatenation of consecutive downstream subregions of CA. The motivations of the proposed research are 4-fold as follows:

- **M1. Precision medicine:** In current clinical practice, physicians mostly depend on binary or visually determined severity of CA disease for decision-making and risk stratification. Patient-specific highly reproducible quantitative measurement would enable providing a better therapy for patients in terms of efficiency, safety, and cost reduction (Leopold & Loscalzo, 2018; Mangion, Gao, Husmeier, Luo, & Berry, 2018).
- **M2. Decision based on quantitative measures:** Geometric model is convenient to quantify important measures in clinical medicine such as volume and boundary surface area of significant features in human body (Frangi, Niessen, & Viergever, 2001; Saito et al., 2005; Kurata et al., 2015; Sumitsuji et al., 2016). For example, the extent and severity of myocardial ischemia can be quantified effectively and efficiently.

- **M3. Construction of optimization models:** Given quantified measures, optimization models can be formulated for decision-making in clinical practice. For example, determination of target for cardiac stem cell therapy (Oettgen, 2006; Segers & Lee, 2008; Shafiq, Jung, & Kim, 2016).
- **M4. Extension to other organs:** The algorithm and the developed CardiacVis program can be applied to other organs in which the hierarchy of vascular network governs the function of organs, such as brain, lung, liver, kidney, and skeletal muscle (West, Brown, & Enquist, 1997; Dawson, 2005; Razavi, Shirani, & Kassab, 2018).

The contributions of this study are 3-fold as follows:

- **C1:** An algorithm to construct the medial axis of coronary artery.
- **C2:** An algorithm to correspond coronary artery and myocardium.
- **C3:** A program CardiacVis implementing the Medial-ABC algorithm.

Figure 1 shows the overview of the scenario where the proposed study may play a critical role. Given a cardiac computed tomography (CT) (I), the mesh models of both CA (II) and myocardium (III) are derived (Lorenz & von Berg, 2006; de Putter, van de Vosse, Gerritsen, Laffargue, & Breeuwer, 2006; Kim, Sohn, & Im, 2019) from the segmentation of CT images. Note that the construction of surface mesh from real-world data is challenging and there were many studies to construct the surface mesh with desired quality or features (Imai, Hiraoka, & Kawaharada, 2014; Moon & Ko, 2018). Given the mesh model of CA, we construct its medial axis (IV) and segment CA using the medial axis (V). We also segment the myocardium corresponding to the segmented CA (VI). The Medial-ABC algorithm includes the research results of the issues IV, V, and VI in the red box of Fig. 1. The CardiacVis program, which implemented the Medial-ABC algorithm, is freely available at Voronoi Diagram Research Center (<http://voronoi.hanyang.ac.kr/software/CardiacVis>).

Figure 2a–f shows a 3D mesh model of heart structure that corresponds to the cardiac CT image in Fig. 1. Human heart consists of four chambers (i.e. two atriums and two ventricles), valves, CA, and proximal ascending aorta. Figure 2a shows pericardial fat surrounding the entire heart structure. Figure 2b shows CA, ascending aorta, LV, right ventricle, and left atrium after the removal of pericardial fat, right atrium, and pulmonary artery from the heart structure. Figure 2c shows LV, aorta, and CA that consists of left CA (LCA) and right CA (RCA), both connected to the aorta. Figure 2d shows LCA, RCA, and LV. Figure 2e shows LV from a different view. We observe that the mesh model of LV is correct in that it is 2-manifold without a boundary and watertight. However, it has topological handles in the location indicated by the red circle. Figure 2f shows the mesh model of LV. The mesh model(s) was created and prepared at Samsung Medical Center, Korea. Hereafter, we refer to it as (**Samsung**) **Model-13** (there are 20 models prepared for experiments; see Section 6 for details). We note here that in order to get a good analysis result, it is desirable to begin the process with a good mesh model that can be obtained by a good meshing algorithm such as Kim et al. (2019). In this paper, italicized fonts denote mesh models of objects, e.g. CA denotes the mesh model of coronary artery CA.

In this paper, it is assumed that a mesh model consists of a single connected component, i.e. each of LV, LCA, and RCA is a connected mesh model. In addition, a mesh is assumed to be watertight, 2-manifold, and free from a self-intersection.

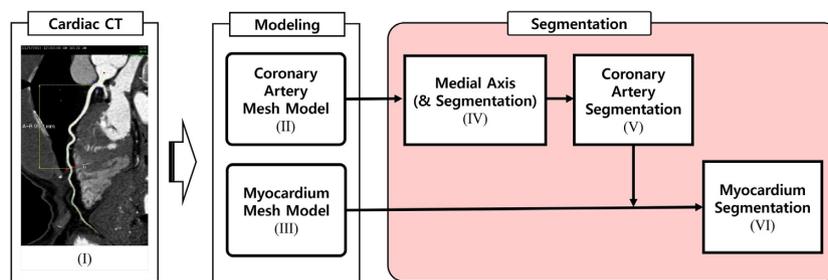


Figure 1: The proposed Medial-ABC algorithm (in the red box) for medicinal diagnosis and treatment. The individual cardiac CT (I) is used to build a 3D triangular mesh model for coronary arteries CA (II) and myocardium (III). Then, the Medial-ABC algorithm constructs the medial axis of the coronary arteries (IV) and computes the segmentations of both coronary artery (V) and myocardium (VI).

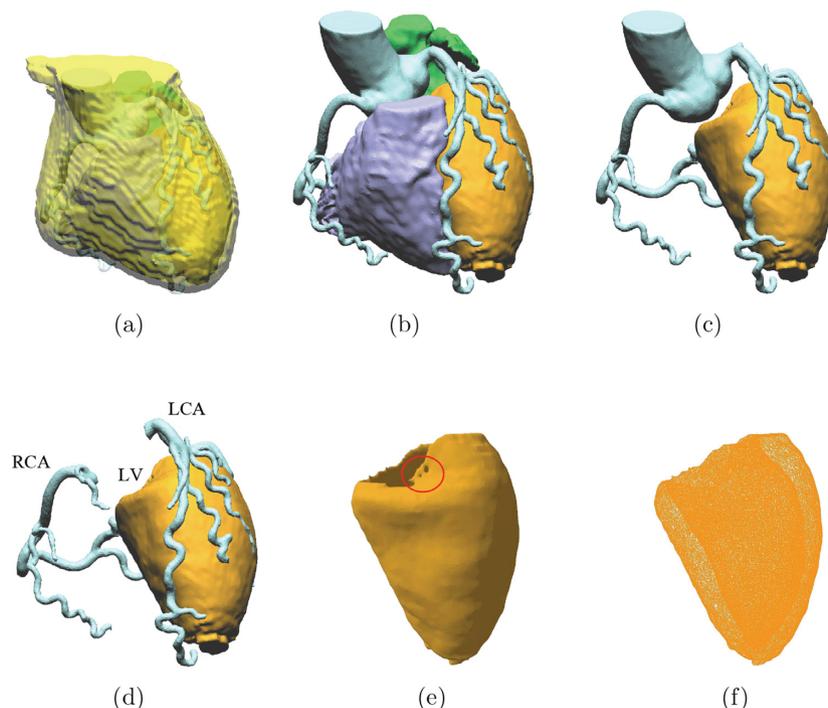


Figure 2: Heart structure (Model-13). (a) Entire heart surrounded by pericardial fat. (b) CA with proximal ascending aorta (sky blue), LV (orange), right ventricle (purple), and left atrium (green). (c) LV, CA with ascending aorta. (d) RCA, LV, and LCA, which play a key role for cardiac function (from left to right). (e) LV viewed from a different orientation (note the holes in the red circle). (f) The mesh model of LV.

The watertightness and 2-manifoldness are tested by the Euler-Poincare formula.

The remainder of this paper is organized as follows. Section 2 reviews related studies. Section 3 presents an algorithm for extracting an adjacency tree from an adjacency graph among the tetrahedral cells of the constrained Delaunay triangulation (CDT) of CA. Section 4 presents an algorithm to construct the medial axis by refining the adjacency tree. Section 5 presents the segmentation of the medial axis, LV, and CA. Section 6 presents experimental result. Section 7 describes the features of the program *CardiacVis* that implemented the Medial-ABC algorithm. Section 8 concludes the paper. Time complexities are worst cases unless otherwise stated.

2. Literature Review

Two technical issues are key to this study: (i) the construction of the medial axis of CA and (ii) the segmentation of LV

and CA using the medial axis that establishes their correspondence. The medial axis (transformation), also called the symmetric axis or skeleton (Kirkpatrick, 1979), was first introduced by Blum in 1967 in order to describe 2D biological shapes (Blum, 1967) and was extensively used for diverse applications such as shape description/matching (Blum & Nagel, 1978; Nackman & Pizer, 1985; Pizer, Oliver, & Bloomberg, 1987), surface reconstruction (Amenta, Bern, & Kamvyselis, 1998; Jalba, Kustra, & Telea, 2013), animation (Wade & Parent, 2002), smoothing or sharpening of shape (Beng Ho & Dyer, 1986), motion planning (Holleman & Kavraki, 2000), and mesh generation (Tam & Armstrong, 1991; Linardakis & Chrisochoides, 2008). In 2D, it is known that the medial axis of planar polygons can be correctly and efficiently constructed using a Voronoi diagram of the polygon (Kirkpatrick, 1979; Lee, 1982) and can be used to compute the offset of the polygon (Kim, Hwang, & Park, 1995; Kim, 1998), packing disks and ellipses (Lee et al., 2018), etc. In 3D, a medial axis can be even used for 3D printing (Wei et al., 2018).

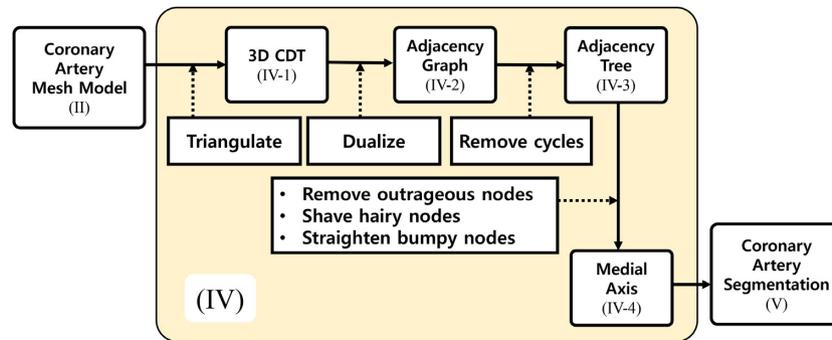


Figure 3: Steps of the Medial-ABC algorithm.

In 3D, however, the construction of a medial surface, the 3D counterpart of the 2D medial axis, remains a challenge due to the following facts: I. (Theory) Medial surface may be composed of both surface patches and degenerate curves (Gong & Bertrand, 1990; Jalba et al., 2013); II. (Computation) Exact computation of medial surface requires heavy computational resources (Culver, Keyser, & Manocha, 2004); and III. (Application) Medial surface is highly sensitive to the noise of input geometry (Shaked & Brucksteiny, 1998; Choi & Seidel, 2004). The technical issues raised by the three viewpoints are all critical. Fact I suggests that a medial surface cannot be conveniently represented in a single mathematical representation. Fact II explains the difficulty of maintaining a correct topology structure of a medial surface during computation due to the algebraic complexity of a medial surface. Note that a correct maintenance of topological structure is the most critical issue in the construction of both medial axis and the Voronoi diagram (Sugihara & Iri, 1989, 1992, 1994; Sugihara, 1992; Yap, 1997; Kim, Cho, Kim, & Kim, 2014). Culver et al. indeed presented a medial axis algorithm for 3D polyhedra using exact arithmetic (together with its implementation) (Culver et al., 2004). However, the algorithm turned out impractical due to the enormous computational requirement of the medial surface even for a small-sized polyhedron with hundreds of faces. Fact III is particularly meaningful for the present study in that our mesh models are derived from CT images that almost always contain measurement error. In addition, 3D medial surfaces usually have many trivial or insignificant parts related with tiny disturbances of polyhedral geometry. Hence, the computation of theoretically correct medial surface is neither feasible nor required nor desirable for the present study and a sufficiently good approximation of medial surface or medial axis is well justified. Hereafter, we will use “medial axis (transformation) \mathcal{M}_{AT} ” to refer to both 2D and 3D unless otherwise necessary.

A common approach to the approximation of a medial axis might be first to compute the Voronoi diagram of points on a model boundary and then to trim off or remove the insignificant parts of the Voronoi diagram (Attali & Montanvert, 1997; Amenta, Choi, & Kolluri, 2001; Dey & Zhao, 2004). This approach seems reasonable as Brandt showed in 2D that the Voronoi vertices inside the shape boundary converges to its medial axis as the sampling rate increases (Brandt, 1994). However, practical consideration of the trade-off between the computational requirement and solution quality due to the number of sampling points on the model boundary becomes a major bottleneck of this approach. Attali and Montanvert proposed an approximation algorithm of a 3D medial axis using the Voronoi diagram of intersection points of 3D spherical balls, which approximates the shape (Attali & Montanvert, 1997).

The segmentation of myocardium and blood vessel was intensively studied. Several image-based studies (Suri, 2000; Mitchell et al., 2001; Paragios, 2003; Lesage, Angelini, Bloch, & Funka-Lea, 2009; Kurata et al., 2015) were reported. In Computer-Aided Design, segmentation were reported based on the features of surface mesh (Xu, Zhou, Wu, Shui, & Ali, 2015; Park, Lee, Chae, & Kwon, 2019). One noteworthy work was to segment the CT image of LV using that of CA: In this work, a user manually picks some voxel points belonging to CA so that each voxel of LV can be assigned to its closest picked point (Debarba, Zanchet, Fracaro, Maciel, & Kalil, 2010; Kurata et al., 2015). Image-based approach was also used for analyzing the morphometry of CA, e.g. the diameter, branching pattern, etc. (Wischgoll, Choy, Ritman, & Kassab, 2008). A notable improvement was made by fusing both image and mesh representations (Termeer et al., 2010): The centerline of CA was computed by CT image with the identification of branch points (Lorenz, Renisch, Schlathöler, & Bülow, 2003) and the centerline was projected to the boundary surface of LV. Then, the Voronoi diagram of the projected points on the boundary of LV was computed with the geodesic distance.

3. Adjacency Tree of Coronary Artery

We want to construct the medial axis (transformation) \mathcal{M}_{AT} of CA as an 1D curve-skeleton (Cornea, Silver, & Min, 2007; Liu, Chambers, Letscher, & Ju, 2010) so that the segmentation of CA and LV can be done by referring to \mathcal{M}_{AT} . We decompose Step IV in Fig. 1 into four substeps as shown in Fig. 3. Section 3 discusses the steps necessary to get an adjacency tree (i.e. Steps IV-1, IV-2, and IV-3) and Section 4 discusses Step IV-4.

3.1. Adjacency graph derived from constrained Delaunay triangulation: Steps IV-1 and IV-2

First, we construct the tetrahedral mesh of CA and use the adjacency information among the tetrahedra. Among many possible triangulations, we found the CDT was most appropriate for the proposed study in that its dual structure contains the desired \mathcal{M}_{AT} while the triangulation is constrained inside of the mesh boundary (Chew, 1989; Aurenhammer, 1991; Okabe, Boots, Sugihara, & Chiu, 1999; Amenta et al., 2001).

Let us look at a 2D example. Consider a simple polygon P in the plane in Fig. 4. Figure 4a and b shows the medial axis of P and the (interior) Voronoi diagram $VD(P)$, respectively. Note that the medial axis is a subset of $VD(P)$ (Kirkpatrick, 1979; Lee, 1982).

This idea, however, cannot be directly applied to the 3D counterpart. The construction of a correct medial axis remains a challenge as was stated by the facts I, II, and III in the previous section. Hence, we instead develop a heuristic algorithm based on

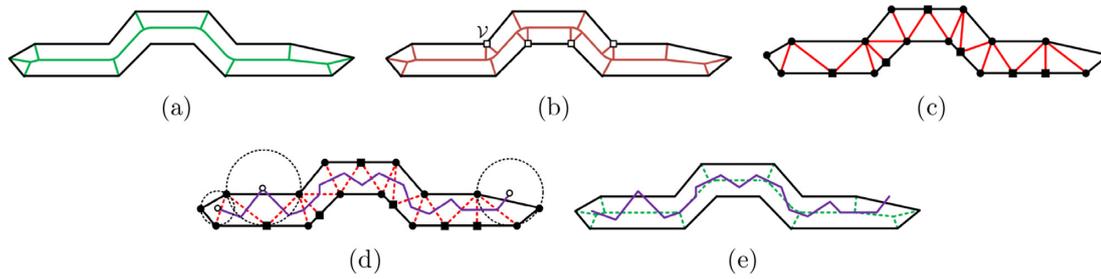


Figure 4: A polygon and its medial axis. (a) The medial axis. (b) The interior Voronoi diagram of the polygon. (c) CDT of points on the polygon boundary. (d) CDT and the corresponding adjacency graph. The tiny white circle denotes the Voronoi vertex and the dashed circle denotes the empty circle of the corresponding Delaunay cell. (e) Both adjacency graph and medial axis.

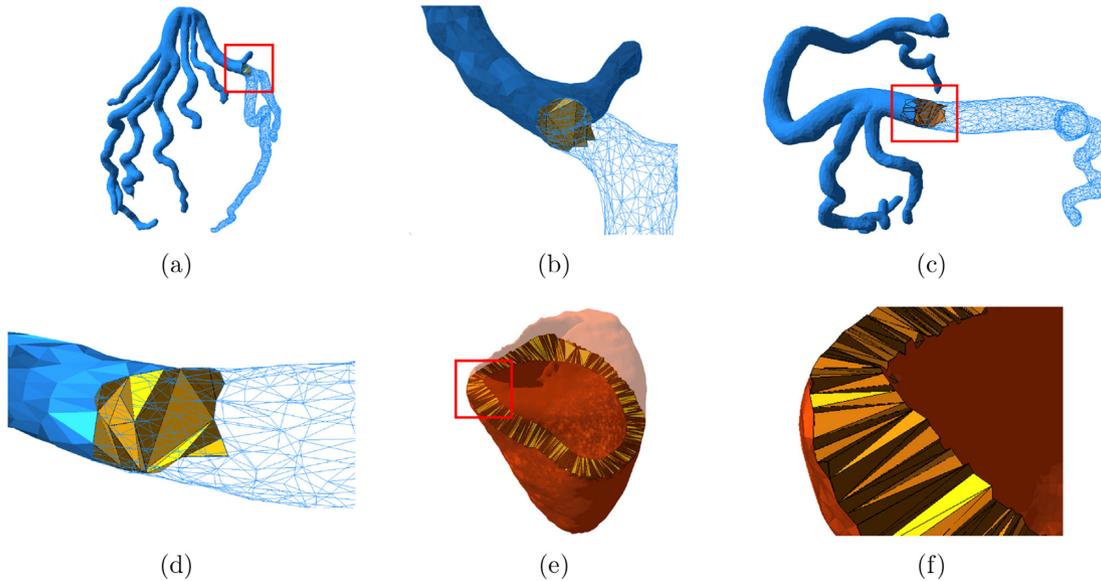


Figure 5: The CDTs of CA and LV of the Model-13. CDTs computed by TetGen Si (2008, 2013, 2015). (a) LCA with some removed CDTs beyond a section plane through the red rectangular window. (b) A close-up of LCA showing the tetrahedral CDT cells around the section plane. (c) RCA with some removed CDTs beyond a section plane through the red rectangular window. (d) A close-up of RCA showing the tetrahedral CDT cells around the section plane. (e) LV with removed CDTs beyond a section plane. (f) A close-up of LV showing the details around the section plane.

CDT. Figure 4c shows the CDT of a set \bar{P} of points on the model boundary, where \bar{P} contains both mesh vertices (i.e. filled circles) and sampled points (i.e. the filled rectangles). Hence, the segment of polygon boundary ∂P becomes one (or more) of the edges in the triangulation. Figure 4d shows a graph that represents the adjacency among the CDT triangles (thus, we call it an adjacency graph; see Definition 1): The tiny white circle denotes a Voronoi vertex, while the dashed circle denotes the empty circle of the corresponding Delaunay cell. The vertices of the graph correspond to the centers of the circumcircles of the triangles. The circumcircles (e.g. those shown as the dotted circles) do not contain any other point generators in the region that is constrained by the polygon boundary, thus leading to the Delaunay property. Note that some circumcircle centers are located outside the model boundary. Figure 4e shows both the adjacency graph and the medial axis. Observe that the adjacency graph relatively well approximates the medial axis. The main idea of this study for the 3D medial axis is based on this simple yet important observation.

We apply the heuristic approach based on the CDT to 3D problems. To construct the 3D mesh model using the CDT structure of the 2-manifold mesh CA, we used the TetGen program (Si, 2008, 2013, 2015) which implemented the algorithm in (Shewchuk, 2003; Si & Gartner, 2011; Si & Shewchuk, 2014). It is

important to check the quality of boundary mesh and repair it if necessary using algorithms such as (Updegrave, Wilson, & Shadden, 2016; Oh, 2019) before constructing the CDT structure. Figure 5 shows the CDTs of CA and LV of Model-13. Figure 5a and b shows LCA and its close-up of one of the sections indicated by the red rectangle, while Fig. 5c and d shows those of RCA. Figure 5e and f shows a section view of LV. The tetrahedral cells with edges are those cut by the section planes.

Definition 1. (Adjacency graph) Consider a triangulation (V, E, F, C) in the 3D space where $V, E, F,$ and C denote the set of vertices, edges, faces, and cells of triangulation, respectively. Let N be the set of nodes where each node $n_i \in N$ one-to-one corresponds to a tetrahedral cell $c_i \in C$ of the triangulation. Let L be the set of links between pair of nodes in N . A link $l_k(n_i, n_j) \in L$ is defined if, and only if, c_i and c_j are adjacent to each other in the triangulation. A graph $G(N, L)$ is called the **adjacency graph** of the triangulation.

The position of $n \in N$ is the center of the circumsphere of the corresponding tetrahedron. We observe the following problems that need to be appropriately handled.

- G may have cycles.
- A node of G may be positioned outside of CA.

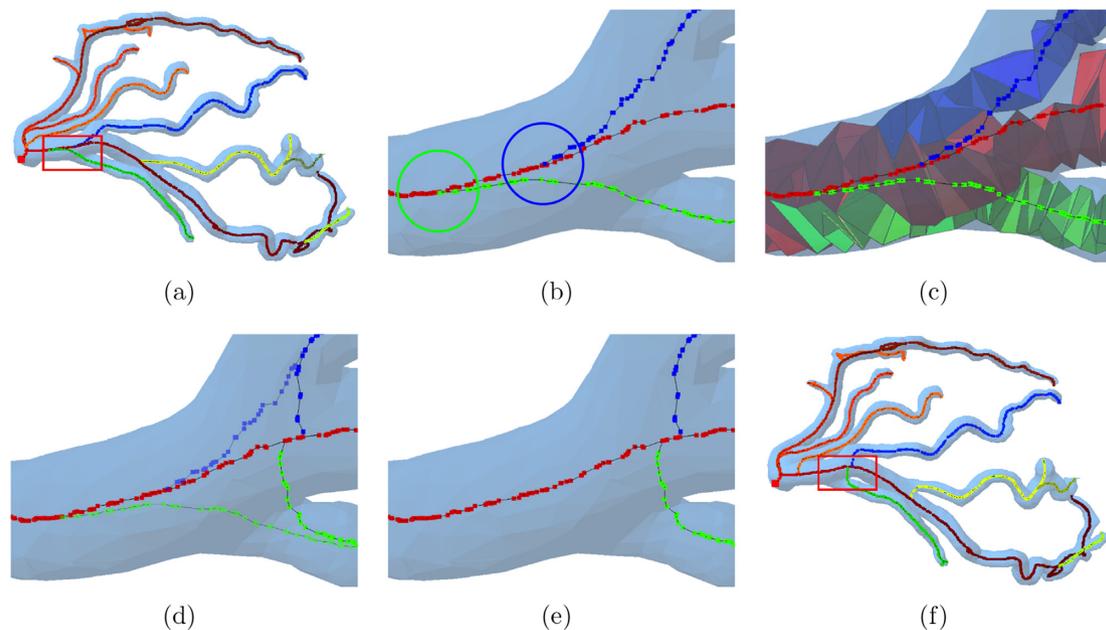


Figure 6: A premature split in the LCA by the forward Dijkstra pass and its removal by the backward Dijkstra pass. (a) LCA mesh model with T^{FWD} . (b) The close-up of a segment: see the two premature splits. (c) The tetrahedral CDT cells corresponding to T^{FWD} . (d) The comparison between T^{FWD} (faded) and T^{BWD} . (e) After the premature split is corrected in T^{BWD} . (f) LCA mesh model with T^{BWD} .

Coronary artery is cycle-free in human body and so should be its medial axis. Cycle-free in mesh model means the model has no handle, i.e. $g = 0$ where g represents the genus of the model. Getting a mesh model CA of genus zero from a CT image may not be easy (see Fig. C1). Obviously a medial axis should exist within the corresponding CA.

Lemma 1. Given a constrained Delaunay triangulation $\text{CDT}(V, E, F, C)$, an adjacency graph $G(N, L)$ can be constructed in $O(|F|)$ time from CDT.

Proof: Suppose that CDT is stored in the simplicial complex data structure (Kim, Kim, Cho, & Sugihara, 2006). Then, a traversal from a cell to its adjacent cell takes $O(1)$ time. As there are $O(|F|)$ adjacency relationships for CDT with $|F|$ triangular faces, the construction of the adjacency graph can be done in $O(|F|)$ time. Note that $|F| = O(|C|)$ where $|C|$ represents the number of tetrahedral cells in CDT. \square

We store $G(N, L)$ in the data structure consisting of two lists for N and L . Each link $l \in L$ has two pointers to its nodes $n_1, n_2 \in N$ and each node n_i has four pointers to up to four incident links in L , which represent the adjacency relationship in CDT.

3.2. Adjacency tree from adjacency graph: Step IV-3

The topology of CA in human body is a tree (i.e. CA has no cycle) and so should be its medial axis. Suppose that the mesh model CA has no handle. An adjacency graph of CA, however, usually has cycles that need to be removed.

Definition 2. (Adjacency tree) A graph $G'(N, L') \subseteq G(N, L)$, $L' \subseteq L$, is an adjacency tree if it has no cycle.

Adjacency tree is not unique in general because there are usually multiple ways to remove a cycle from a graph. We use the Dijkstra algorithm (Dijkstra 1959) to reduce an adjacency graph

to an adjacency tree with a manually selected root node. We call it the **forward Dijkstra pass**.

3.2.1. Premature splits and phantom splits by a forward Dijkstra pass

A forward Dijkstra pass alone may not be able to solve the problem completely. The constructed adjacency tree may be different from the ideal medial axis in that the branch nodes of the adjacency tree tend to be placed closer to the root node than it is in the ideal medial axis. We call this phenomenon a **premature split** of branches.

Figure 6a shows an example of premature split that occurred in the LCA of Model-13 by the forward Dijkstra pass. Note in the close-up in Fig. 6b that the adjacency tree branches at a node much closer to the root node than at the node where the medial axis of CA actually branches. Figure 6c shows the tetrahedral cells corresponding to the two branches. A premature split in this example causes two branches of the adjacency tree run in parallel for some length, which can be very long depending on the arrangement of tetrahedral cells of CDT. Hence, a premature split exists, the correspondence between CA and LV can be less accurate, and this could cause a serious miscalculation of correspondence. Figure 6d–f shows the adjacency tree after the premature split is fixed (using the backward Dijkstra pass that is explained below).

An even more critical problem that can be caused by a forward Dijkstra pass is a nonsense **phantom split** constructed in CA, which should not exist. Figure 7a shows RCA and the corresponding adjacency tree after the forward Dijkstra pass. Note the region of RCA within the black rectangle. Obviously, we expect one branch of medial axis because the artery is a tube-like shape around the region. However, its close-up in Fig. 7b clearly shows two branches: one from the main stream of the artery (in blue) and an additional relatively short branch (in red) from a consecutive trivial tetrahedra in the same mesh model. The second is nonsense and should be removed. Figure 7c shows the

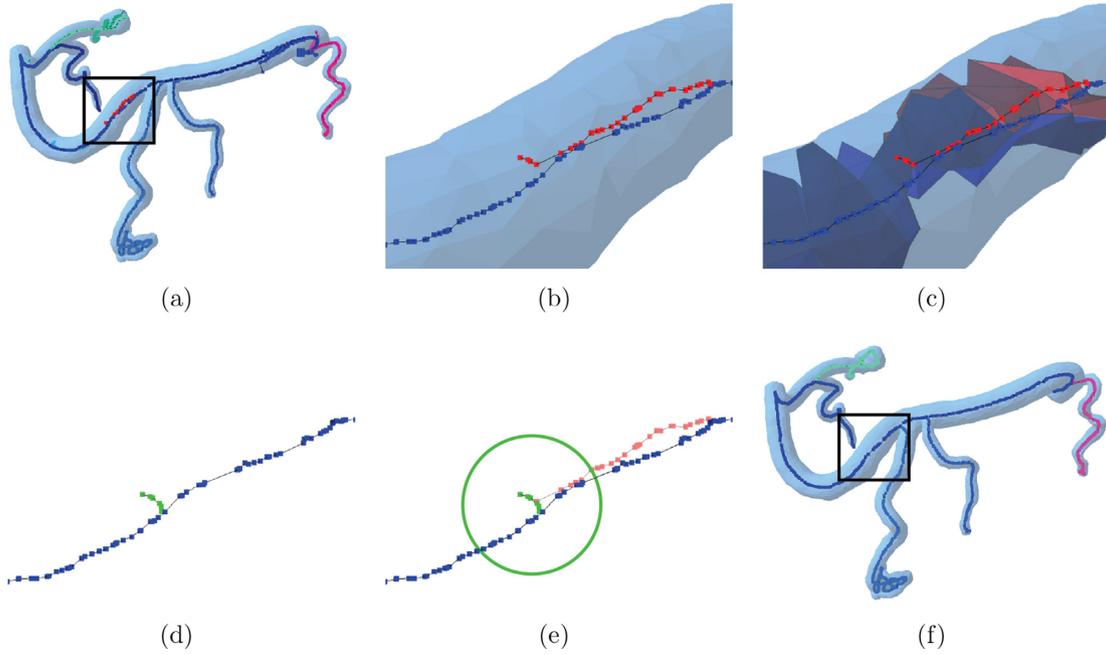


Figure 7: A nonsense phantom split/branch in RCA by the forward Dijkstra pass and its removal by the backward Dijkstra pass. (a) RCA and the adjacency tree produced by the forward Dijkstra pass. (b) A close-up showing the nonsense phantom branch (in red) split from the main branch (in blue). (c) The CDT cells corresponding to the nodes on the branches. (d) The phantom branch trimmed off to a hairy node (in green) by the backward Dijkstra pass. (e) Both hairy node and phantom branch shown together. (f) RCA and the adjacency tree after all phantom branches are removed.

tetrahedral cells corresponding to the two branches of the adjacency tree.

3.2.2. Remedy of both premature splits and phantom splits

In order to detect and remedy these abnormal splits, we employ another pass of the Dijkstra algorithm called the **backward Dijkstra pass**. The idea is to modify the information produced by the forward Dijkstra pass with the information obtained by the backward pass.

Definition 3. (Forward shortest path tree) Let $\pi_i^{\text{FWD}}(N_i^{\text{FWD}}, L_i^{\text{FWD}})$ be the forward shortest path from the root node to the node n_i in G . Let $T^{\text{FWD}}(N^{\text{FWD}}, L^{\text{FWD}}) \subseteq G(N, L)$ be the **forward shortest path tree** where $N^{\text{FWD}} \equiv N$ and $L^{\text{FWD}} = \bigcup L_i^{\text{FWD}}$.

Lemma 2. T^{FWD} can be constructed in $O(|N|\log|N|)$ time by applying the forward Dijkstra pass to G .

Hereafter, we refer to a forward shortest path as a **forward path**. We define the length of a path as the sum of the lengths of all links in the path. We use T^{FWD} to construct T^{BWD} , the **backward shortest path tree**, by the backward Dijkstra pass as follows.

Definition 4. (Backward shortest path) Let $T(N^T, L^T) \subseteq G(N, L)$ be a tree. Let n_i be a leaf node of T^{FWD} . A connected path from n_i to a node $n_j \in N^T$ is called a **backward candidate path** of n_i to T . The **backward shortest path** from n_i to T is the backward candidate path whose length is minimal. Hereafter, we refer backward shortest path as **backward path**.

Definition 5. (Concatenation) Let $T(N^T, L^T) \subseteq G$ be a tree and $\pi(N^\pi, L^\pi) \subseteq G$ a path. **Concatenation** $\pi \oplus T$ is defined as $T(N', L')$ where $N' = N^T \cup N^\pi$ and $L' = L^T \cup L^\pi$.

Suppose that $T_{i-1}^{\text{BWD}} \subseteq G(N, L)$ is available. Let π_i^{BWD} be the backward path from the leaf node n_i of T^{FWD} to T_{i-1}^{BWD} . Then, we get T_i^{BWD} by concatenating π_i^{BWD} to T_{i-1}^{BWD} . The backward shortest path tree T^{BWD} is defined as follows.

Definition 6. (The backward shortest path tree) The backward shortest path tree T_n^{BWD} with n nodes is

$$\begin{aligned} T_n^{\text{BWD}} &= \pi_n^{\text{BWD}} \oplus T_{n-1}^{\text{BWD}} = \pi_n^{\text{BWD}} \oplus \pi_{n-1}^{\text{BWD}} \oplus T_{n-2}^{\text{BWD}} \\ &\dots = \pi_n^{\text{BWD}} \oplus \dots \oplus \pi_2^{\text{BWD}} \oplus \pi_1^{\text{BWD}}. \end{aligned} \quad (1)$$

We represent both T^{FWD} and T^{BWD} in the same data structure used for $G(N, L)$ but, at each node, with a refined interpretation to distinguish the parent from siblings. Note that we keep these three constructs as distinct objects in distinct storage. We additionally maintain an ordered queue Q^{FWD} , which stores all forward paths in T^{FWD} in the non-increasing order of path length. Hence, the first element of Q^{FWD} corresponds to the longest path in T^{FWD} . Each element of Q^{FWD} has the information about the corresponding leaf node in T^{FWD} at which the path ends.

Figure 8 shows a schematic diagram of an example of a backward shortest path tree construction process through the backward Dijkstra pass (i.e. concatenating the backward paths generated by the Dijkstra algorithm). The blue polygons denote ∂CA and the solid chains denote forward paths. Suppose that Fig. 8a shows T^{FWD} with seven nodes: one root node (shown as the filled black circle) and six leaf nodes (shown as unfilled white circles). Note the premature splits (e.g. n_3 and n_4), phantom splits (e.g. n_2 and n_6), and the node positioned outside of ∂CA (e.g. n_5). In the figure, the node indices correspond to their locations of the forward path π^{FWD} in Q^{FWD} . For example, π_1^{FWD} and π_2^{FWD} are the first and second longest in T^{FWD} and thus are the first and second elements in Q^{FWD} .

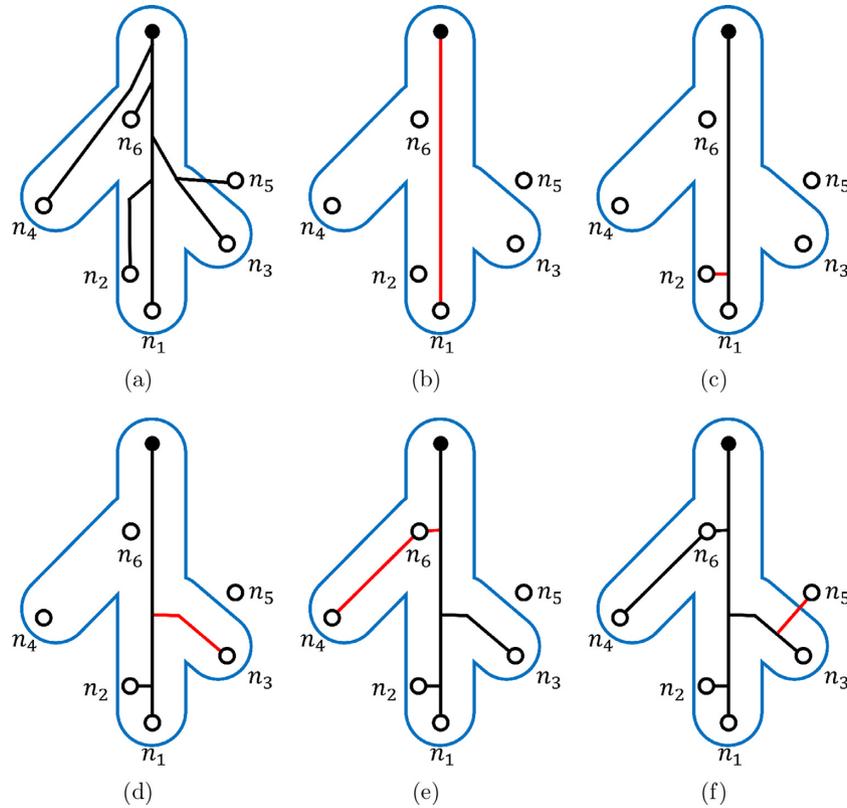


Figure 8: The backward Dijkstra pass to remedy both premature and phantom splits caused by the forward Dijkstra pass. (a) T^{FWD} . (b) T_1^{BWD} that consists of a single longest forward path π_1^{BWD} (i.e. the red path from n_1 to the root node; the initial T^{BWD}). (c) T_2^{BWD} by concatenating π_2^{BWD} to T_1^{BWD} . (d) T_3^{BWD} by concatenating π_3^{BWD} to T_2^{BWD} . (e) T_4^{BWD} by concatenating π_4^{BWD} to T_3^{BWD} . (f) The final backward tree T_5^{BWD} obtained by concatenating π_6^{BWD} .

The backward Dijkstra pass is as follows. We remove the first element of Q^{FWD} , which is π_1^{FWD} . Note that this is the longest forward path of T^{FWD} (the red chain in Fig. 8b). This forward path is identical to the backward path from n_1 to the root node n_{root} of T^{FWD} . Hence, $\pi_1^{\text{BWD}} \equiv \pi_1^{\text{FWD}}$. Let $T_1^{\text{BWD}} = \pi_1^{\text{BWD}}$. Figure 8b shows T_1^{BWD} .

We remove the next element of Q^{FWD} : It corresponds to π_2^{FWD} , which is the second longest forward path of T^{FWD} . Here, we apply the backward Dijkstra algorithm to compute the backward path π_2^{BWD} from n_2 to T_1 . The red chain from n_2 to T_1 in Fig. 8c denotes π_2^{BWD} . We concatenate π_2^{BWD} to T_1^{BWD} to construct T_2^{BWD} as shown in Fig. 8c.

We remove the next element of Q^{FWD} and compute the backward path π_3^{BWD} from n_3 to T_2^{BWD} to produce T_3^{BWD} (the red chain in Fig. 8d). We repeat this process (as shown in Fig. 8e and f) until Q^{FWD} is empty.

Note that it is possible that a node of distinct forward path gets included in another backward path. For example, n_6 is one of the internal nodes of π_4^{BWD} as shown in Fig. 8e. Therefore, $\pi_6^{\text{BWD}} = \emptyset$ and n_6 is not a leaf node of T^{BWD} anymore. Hence, T^{BWD} may have fewer nodes than T^{FWD} has. Figure 8f shows the final T^{BWD} constructed by the backward Dijkstra pass that remedies both problems of T^{FWD} .

Observation 1. T_n^{BWD} has neither premature split nor phantom split.

Algorithm 1 computes an adjacency tree with neither premature nor phantom split by the forward and backward Dijkstra passes. Lines 1 through 6 compute T^{FWD} and store all paths in

the ordered queue Q^{FWD} . Line 7 constructs the initial adjacency tree T_1^{BWD} . Lines 8 through 14 iteratively update the subtree by concatenating a backward path from a leaf node to the intermediate backward tree. Note that line 11 implies that the length of a backward path π^{BWD} can be zero when the leaf node is already included in the intermediate backward tree (as shown in Fig. 8e). The premature splits in Fig. 6 and the phantom splits in Fig. 7 are fixed by Algorithm 1.

The Dijkstra algorithm takes $O(|L| + |N|\log|N|)$ time for $|N|$ nodes and $|L|$ links in $G(N, L)$ (Fredman & Tarjan, 1984). Algorithm 1 has two passes of Dijkstra: forward and backward. Forward Dijkstra pass computes the shortest paths of all nodes from the root and backward pass consists of the runs of Dijkstra algorithms for all leaf nodes. We note that a backward tree construction is most time consuming in the proposed algorithm as stated by the following lemma.

Lemma 3. T^{BWD} can be constructed from $G(N, L)$ in $O(|N|^2)$ time.

Proof: Construction of T^{FWD} from G takes $O(|N|\log|N|)$ time. Suppose that T^{FWD} has m leaf nodes. Let $\pi_i^{\text{BWD}} = \pi_i^{\text{BWD}}(N_i, L_i)$ be the i -th backward path. Computing π_i^{BWD} takes $O(|N_i|\log|N_i|)$ time. Hence, computing π_1^{BWD} through π_m^{BWD} takes $O(\sum |N_i|\log|N_i|)$. Because $\log|N_i| \leq |N_i|$, $O(\sum |N_i|\log|N_i|) \leq O(\sum |N_i|^2)$. Because $\sum |N_i|^2 \leq (\sum |N_i|)^2$ and $\sum |N_i| = O(|N|)$, $O(\sum |N_i|^2) \leq O(|N|^2)$. Therefore, construction of T^{BWD} from G takes $O(|N|^2)$ time. We believe a tighter bound exists. \square

Algorithm 1: Adjacency Tree from Adjacency Graph

input : Adjacency graph G , root node n_{root} of G
output: Adjacency tree T^{BWD}

- 1 Construct the forward shortest path tree T^{FWD} of G with n_{root} .
- 2 **for** a leaf node n_i in T^{FWD} **do**
- 3 Compute each forward path π_i^{FWD} from n_{root} to n_i .
- 4 Compute the path length of π_i^{FWD} .
- 5 Insert n_i and π_i^{FWD} into the ordered queue Q^{FWD} according to the non-increasing order of path length.
- 6 **end**
- 7 Construct the initial adjacency tree T_1^{BWD} .
- 8 **while** Q^{FWD} is not empty **do**
- 9 Remove the first element n_i^Q of Q^{FWD} .
- 10 Compute the backward path π_i^{BWD} from a leaf node n_i to T_{i-1}^{BWD} (n_i corresponds to n_i^Q).
- 11 **if** the path length $|\pi_i^{BWD}| > 0$ **then**
- 12 Concatenate (i.e. $T_i^{BWD} = \pi_i^{BWD} \oplus T_{i-1}^{BWD}$).
- 13 **end**
- 14 **end**

4. Medial Axis from Adjacency Tree: Step IV-4

T^{BWD} after the backward Dijkstra pass is an initial solution for constructing medial axis. T^{BWD} is cycle-free but it may still possess three issues that need to be handled: outrageous nodes, hairy nodes, and bumpy nodes. In Fig. 8f, n_5 is outrageous and n_2 is hairy. Figure 9 shows the process to construct the \mathcal{MAT} from the adjacency tree. Figure 9a and b shows T^{FWD} and T^{BWD} , respectively. Figure 9c shows T^{BWD} after removing the nodes out-

side of CA. Figure 9d shows the adjacency tree after removing the nodes with relatively insignificant contribution to \mathcal{MAT} . Figure 9e shows the remaining paths after straightening: We consider T^{BWD} in Fig. 9e the \mathcal{MAT} . These steps are referred to as “outrageous node removal,” “hairy node shaving,” and “bumpy node straightening,” respectively. This section presents the process to construct \mathcal{MAT} by improving the initial solution T^{BWD} by appropriately handling these issues.

4.1. Outrageous node removal

Medial axis should be contained within the boundary of mesh model. However, T^{BWD} may have nodes that are placed outside of CA. Recall that the position of a node is the center of a circumsphere of a tetrahedral cell and thus it can be positioned outside of model boundary. We call such a node *outrageous*. Figure 10a shows the outrageous nodes of T^{BWD} . Figure 10b and c is close-ups.

Definition 7. (*Outrageous node*) A node of T^{BWD} is *outrageous* if it is located outside of CA.

Hence, the removal of outrageous nodes improves the quality of T^{BWD} . It turns out that an outrageous cell tends to be relatively flat and located near ∂CA (Amenta et al., 2001). Figure 10b shows some (blue) outrageous nodes and the corresponding (red) CDT cells near ∂CA . An outrageous node can be easily identified by testing the existence of an intersection between its incident links and ∂CA . An outrageous node can be either a leaf node or non-leaf node in T^{BWD} .

Lemma 4. *Outrageous nodes of $T^{BWD}(N, L)$ can be removed in $O(|N|)$ time on average.*

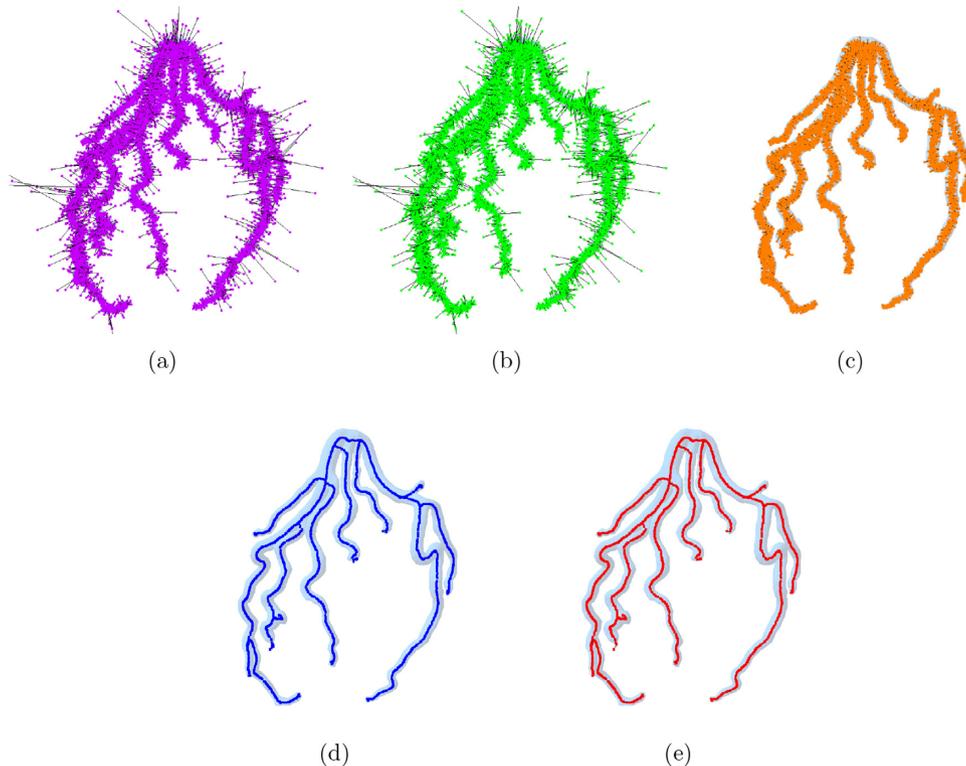


Figure 9: \mathcal{MAT} from T^{BWD} in LCA. (a) T^{FWD} . (b) T^{BWD} . (c) After outrageous node removal. (d) After hairy node shaving. (e) After bumpy nodes straightening.

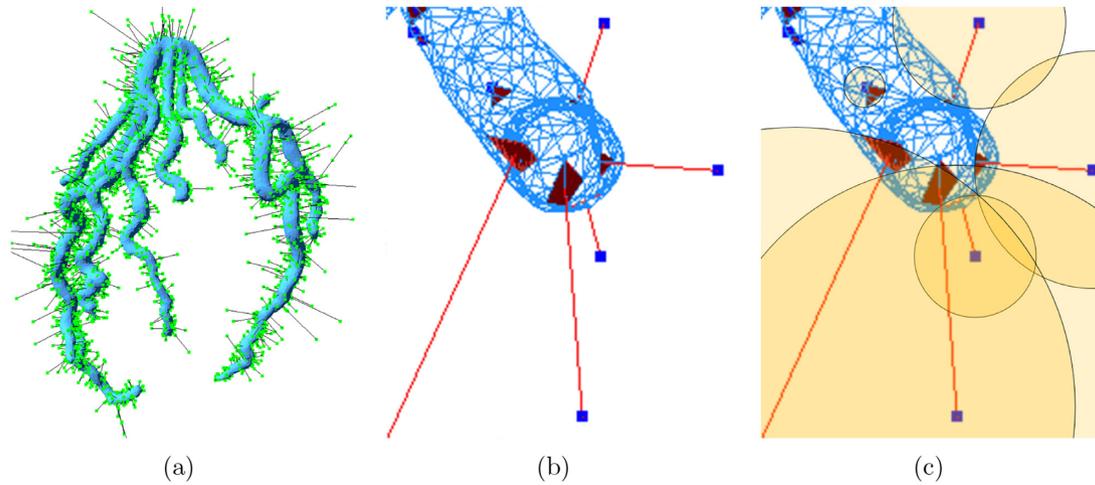


Figure 10: Example of the outrageous nodes in T^{BWD} of LCA. (a) All outrageous nodes shown with the mesh model. (b) Close-up of an artery tip showing a few outrageous nodes. (c) Circumspheres of the outrageous cells.

Proof: Given $CA(V, E, F)$, we need to perform the intersection test between the line segment corresponding to $l \in L$ and a triangular face $f \in F$. If we do not use any acceleration, it would take $O(|L||F|)$ time. With a spatial bucket that has $O(|F|)$ elements for acceleration, it takes $O(1)$ time on average for detecting an intersection and removing the outrageous node. Hence, it takes $O(|L|)$ time on average for removing all outrageous nodes. Recall that $|L| \leq O(|N|)$.

4.2. Hairy node shaving

Even after outrageous nodes are removed, T^{BWD} usually contains tiny subtrees that are meaningless for being a part of medial axes. Figure 11a shows T^{BWD} of LCA after the outrageous nodes are removed. Figure 11b is a close-up of the region in the red box that shows several tiny subtrees after removing outrageous nodes. We call such a subtree a **hair** because of its tiny and noisy contribution to medial axis and the operation to remove hair is called **shaving**. Figure 11c and d shows the hair and the (orange) CDT cells corresponding to the hair. In Fig. 11e, the blue color represents T^{BWD} , together with their corresponding CDT cells, after shaving the hairy nodes. Compare the blue cells with the orange cells.

The idea of shaving is straightforward. Let $\pi_i \subset T^{\text{BWD}}$ be the path from a leaf node n_i to the root node of T^{BWD} . We measure the contribution of each path π_i to T^{BWD} and shave (or ignore) it off from T^{BWD} if its contribution is relatively trivial. The specific shaving process can be found in Appendix A.

Lemma 5. *Hairy node shaving of $T^{\text{BWD}}(N, L)$ takes $O(|N|)$ time.*

Proof: Suppose that T^{BWD} has m leaf nodes. First, we compute the distance from the root to each node of T^{BWD} in $O(|N|)$ time. When $\pi_i(N_i, L_i)$ is concatenated to T_{i-1} , the length λ of other remaining paths in T^{BWD} needs to be updated. We can update λ in $O(1)$ time by referring to the distance attribute. As there can be $O(|N|)$ paths, this process can be done in $O(|N|)$ time. \square

4.3. Bumpy node straightening

T^{Shaved} usually represents the topology of CA pretty well. However, it may contain some **bumpy nodes** that are eccentric from

the medial axis of CA. The bumpy nodes can lower the quality of the mapping between LV and CA. In this section, we present how to straighten bumpy nodes so that T^{Shaved} fits better for a medial axis.

Consider three consecutive nodes n_{i-1}, n_i, n_{i+1} of a shaved adjacency tree. Let \vec{v}_i be the vector from n_{i-1} to n_i . Let $\theta_i < 180^\circ$ be the angle between \vec{v}_i and \vec{v}_{i+1} at n_i (this is called the **nodal angle** of n_i). We consider θ_i the measure of the turn of the path at n_i and measure it at every node with degree two in T^{Shaved} . If the nodal angle of n_i is greater than a threshold, say θ_{\max} , we consider it a bumpy node and remove. More specific discussion to determine θ_{\max} can be found in Appendix B.

To find bumpy nodes, we need to check the nodal angle of all nodes in T^{Shaved} , thus leading to the following time complexity.

Lemma 6. *Bumpy node straightening of $T^{\text{Shaved}}(N, L)$ can be done in $O(|N|)$ time.*

Theorem 7. *\mathcal{MAT} can be constructed for $CA(V, E, F)$ in $O(|V|^2)$ time.*

Proof: From the lemmas above, the most time-consuming process for computing \mathcal{MAT} is the backward Dijkstra pass that takes $O(|N|^2)$ time where $|N| = O(|V|)$ for $G(N, V)$. \square

5. Segmentations of Medial Axis, Arteries, and Ventricles

For the diagnosis and decision-making of treatment, the localization and evaluation of the extent of regional myocardium at risk subtended by obstructed CA is critical. For this purpose, we segment the geometric models of \mathcal{MAT} , coronary artery CA, and left ventricle LV (i.e. Steps V and VI in Fig. 1).

5.1. Segmentation of medial axis

First of all, it is necessary to segment a medial axis according to a physician's need. The basic segmentation method is an automatic mode to split a medial axis into pieces at branch nodes with the degree three. A physician may want to find the relationship between a smaller segment of artery and heart muscle. In this case, a medial axis can be segmented by a manual mode.

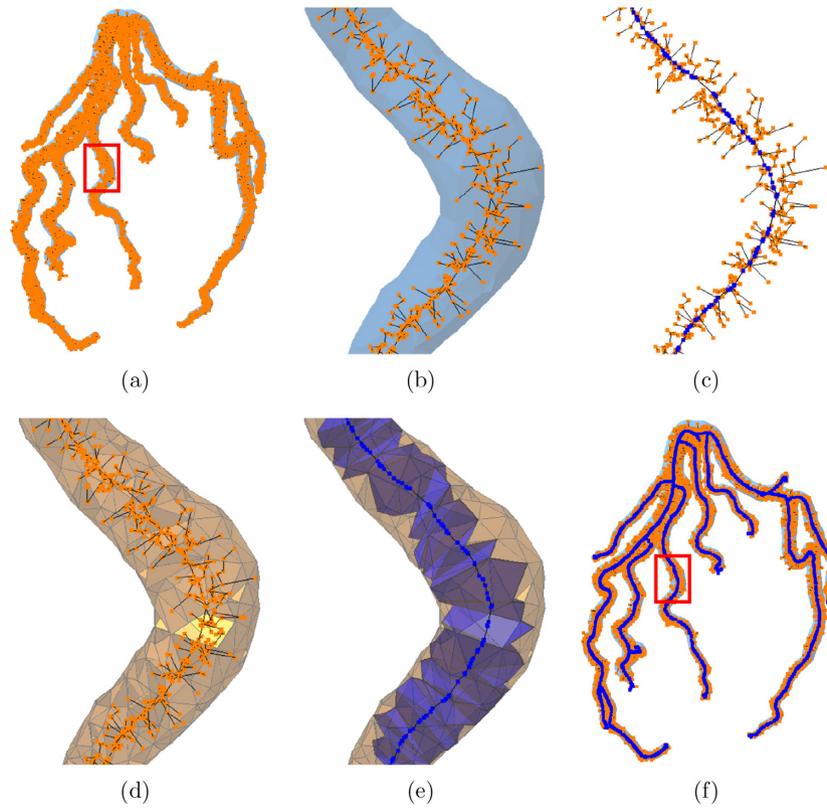


Figure 11: Hairy node shaving in LCA. (a) T^{BWD} with hairy nodes (outrageous nodes are removed). (b) Close-up of a CA segment (CDT cells and T^{BWD}). (c) T^{BWD} (orange) and T^{Shaved} (blue). (d) CDT cells of the hairy nodes. (e) T^{Shaved} and the corresponding CDT cells. (f) Overall structure of T^{BWD} and T^{Shaved} .

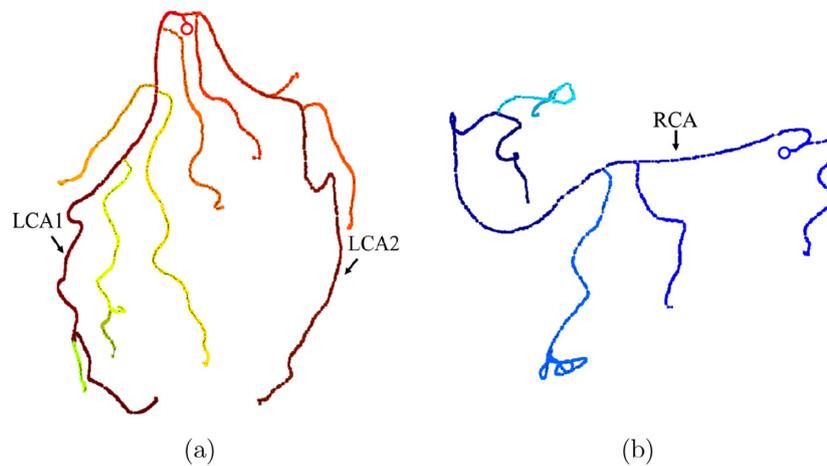


Figure 12: Segmentation of the MAT of CA. (a) MAT of the LCA. (b) MAT of the RCA. In both figures, hollow red and blue circles denote the root node.

We consider MAT as a set of segments, i.e. $MAT = \{Mat_1, Mat_2, \dots\}$. $Mat_i \in MAT$ consists of an ordered set of one or more node. Figure 12a and b shows the segmentations of medial axes of LCA and RCA, respectively. The branch nodes can be found to segment MAT by scanning the nodes, thus leading to the following lemma.

Lemma 8. Given $MAT = MAT(N, L)$, segmentations of MAT takes $O(|N|)$ time.

5.2. Segmentation of coronary artery

We want to segment CA according to the segmentation of MAT by finding the portion of CA corresponding to each segment of MAT . Let CDT^{CA} be the constrained Delaunay triangulation CDT of CA. Then, we formulate the segmentation of CA as an optimization problem called an assignment problem: Assign each CDT cell of C to one, and only one, node of N in MAT . The measure of the optimization problem is the total distance between each pair of cell and node. We use the Euclidean distance be-

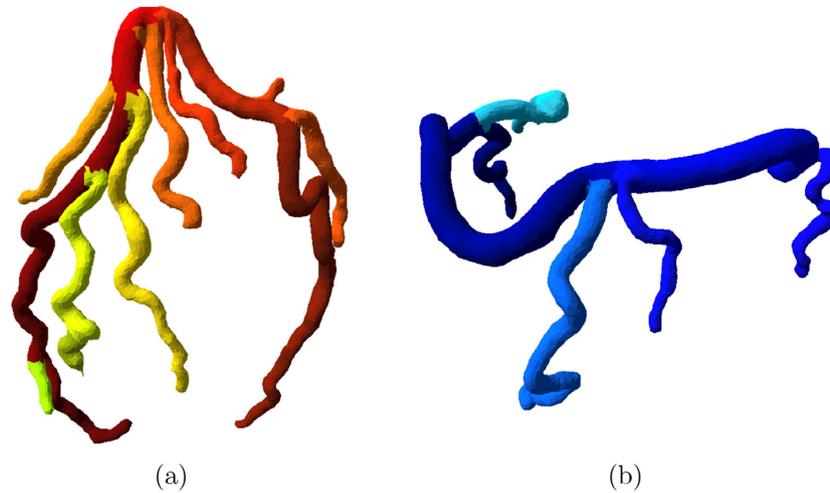


Figure 13: Segmentation of the coronary artery CA using the segmented medial axis. (a) The segmentation of the LCA. (b) The segmentation of the RCA.

tween the mass center of tetrahedral CDT cell and the position of the node on MAT . Recall that a node position is the center of the circumsphere of a tetrahedral cell and the center of mass is located within tetrahedron.

The segmentation problem, referred to as CA-SEG-OPT, is therefore a many-to-one assignment problem and can be formulated as an integer linear program as follows:

$$\text{Minimize } \sum_{i \in I} \sum_{j \in J} d(c_i, n_j) x_{ij}, \quad (2)$$

$$\text{s.t. } \sum_{j \in J} x_{ij} = 1, i \in I, \quad (3)$$

$$x_{ij} \in \{0, 1\}, \quad (4)$$

where I and J are the indices of the sets for C and N , respectively, and $d(c_i, n_j)$ is the Euclidean distance, between the position of $n_j \in N$ and the center of $c_i \in C$. Equation (3) enforces each cell to be assigned to one and only one node. The formulation above can be solved by assigning each CDT cell to its closest node on MAT . Solving the formulation above in a naive way takes $O(|C||N|)$ time in both worst and average case because the assignment of each cell to the nearest node on the medial axis can be done by scanning the distance from the cell to all nodes in N . This computational requirement can be prohibitive for physician to use in practice because both $|C|$ and $|N|$ are usually big numbers. For the example of Model-13, LCA has 2,787 nodes on MAT and 37,999 CDT cells, and RCA has 1,773 nodes on MAT and 24,210 CDT cells. Hence, we have devised a heuristic algorithm that runs fast yet produces an excellent solution.

The heuristic algorithm for the assignment problem above runs as follows. First, we make an initial assignment of cells to the nodes on MAT . Recall that each node on MAT already has a corresponding CDT cell. We use this correspondence directly for the initial assignment taking $O(1)$ time for each assignment of a cell and $O(|N|)$ time for all nodes in N .

Second, we propagate to the neighbor not-yet-assigned cells in CDT. There are many unassigned CDT cells corresponding to outrageous nodes, hairy nodes, and bumpy nodes as shown in Fig. 11.

The propagation is done node by node on MAT in parallel. Suppose that a node n_i is associated with a CDT cell c_i . Be aware that n_i is associated with only one cell after the initial assignment and becomes associated with more cells as the propagation proceeds. Let c_j be one of the CDT cells adjacent to c_i that is not associated with any node yet. Then, we blindly associate

it to n_i . We repeat this process of one new assignment at a time for all nodes until all cells are assigned. We will eventually get a segmentation of CA where two segments encounter each other at a segment boundary.

Third, we resolve conflicts in the topological assignment. We do not use any measure of distance in the assignment above and therefore the assignment does not necessarily reflect the Euclidean distance. This property may introduce a conflict between the assignment of the cells for two segments in the neighbor. In other words, one cell, say c_k , of the segment Mat_i at the boundary with the segment Mat_j may be closer to the segment Mat_j from Euclidean distance point of view. In this case, we reassign c_k from Mat_i to Mat_j . Both assignment and reassignment of cells to segments take $O(|C|)$ time for all cells in C .

Note the following properties: (i) Segmentation is dependent on propagation sequence and (ii) Segment can be disconnected. Figure 13a and b shows the segmentation of the LCA from two different views.

Lemma 9. Given MAT , $CA(V, E, F)$ can be segmented in $O(|V|)$ time.

Proof: An initial assignment can be done in $O(1)$ using the correspondence between the nodes in MAT and the cells in CDT. With an initial assignment, both assignment and reassignment of cells can be done in $O(|C|)$ time, where $|C|$ denotes the number of cells in CDT. As $|C| = O(|V|)$, CA can be segmented in $O(|V|)$ time. \square

5.3. Segmentation of left ventricle

Left ventricle LV is most critical for the mechanical function of heart and functions with the blood supply by coronary artery CA. As CA supplies blood to LV muscles through perfusion process, we assume the Euclidean distance between the boundaries of CA and LV is fundamental for understanding the properties of the system. In this regard, we want to segment LV into a set of subregions so that each subregion could be assigned to one segment of MAT in proximity.

We treat the segmentation of LV in a similar way to the segmentation of CA in that we use the same formulation of an assignment between the nodes on MAT and the CDT cells within LV. This formulation, referred to as LV-SEG-OPT, is based on the

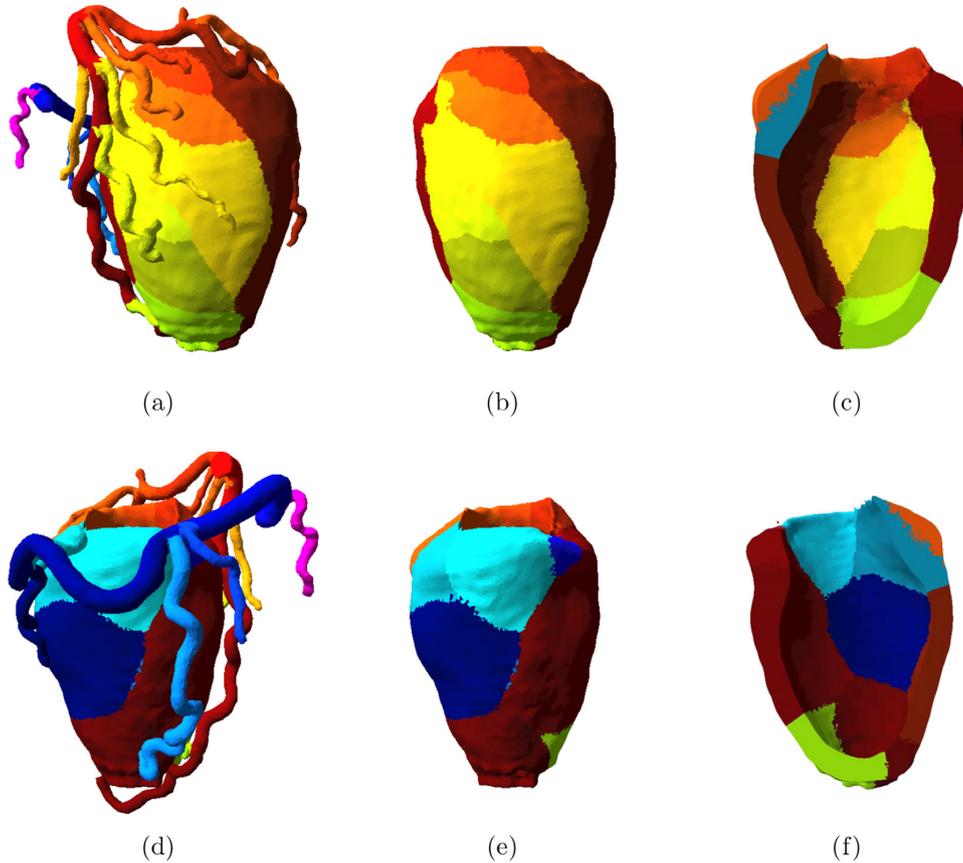


Figure 14: Segmentation of the mesh models of the left ventricle LV and coronary artery CA using the medial axis of CA . (a) The segmentation of LV and CA in LCA side. (b) LV only. (c) LV view from inside. (d) The segmentation of LV and CA view from RCA side. (e) LV only. (f) LV view from inside.

rationale that the closer the CA and LV are, the easier would be the blood supply through microvascular network from CA to LV .

There is, however, an important difference between the two optimization problems. Let CDT^{LV} be the CDT of LV and CDT^{CA} be the CDT of CA , respectively. Recall that in CA -SEG-OPT, there exists an association between the nodes on \mathcal{MAT} and the cells in CDT^{CA} so that the initial assignment for the segmentation can be easily done. In LV -SEG-OPT, however, no such an association exists between the nodes on \mathcal{MAT} and the cells in CDT^{LV} . Hence, it is necessary to establish an association between each node on \mathcal{MAT} and one of the cells in CDT^{LV} . Once the initialization is done, the rest of the process for LV -SEG-OPT is exactly identical to that of CA -SEG-OPT.

A naive approach for solving LV -SEG-OPT also takes $O(|C||N|)$ time, where $|C|$ denotes the number of CDT cells in CDT^{LV} . Note that the number of cells in CDT^{LV} is even bigger than in CDT^{CA} . For the example of Model-13, CDT^{LV} has 202,183 cells. Hence, computational requirement is even more prohibitive than before. We have devised an efficient heuristic for the initial assignment.

Let P^{LV} be the set of the vertices of ∂LV and P^{MAT} the set of the nodes on \mathcal{MAT} . Let $P = P^{LV} \cup P^{MAT}$. Let $DT(P)$ be the Delaunay triangulation of P . Then, $DT(P)$ has the proximity information among the vertices of ∂LV and the nodes on \mathcal{MAT} in a compact representation of the ordinary Voronoi diagram. We use $DT(P)$ to quickly locate the CDT cell that is the closest to each node on \mathcal{MAT} . Note that $DT(P)$ can be computed in $O(|P|\log|P|)$ in the worst case and in most cases, experiments show a strongly linear time behavior to $|P|$.

The traversal in $DT(P)$ to find an initial assignment between a node and a cell is done as follows. Consider a node $p \in P^{MAT}$. We locate a DT cell c^{DT} incident to p and check if there exist any vertex $v \in P^{LV}$ on ∂c^{DT} . If there exists such a vertex v , we check the other DT cells in the neighbor that has another vertex closer than v from p . If such a vertex is not found in the first shell of cells incident to p , we go farther to the second shell of cells. We can eventually find a DT cell c incident to p where c has at least one DT vertex, say q , belonging to P^{LV} . In this way, we can find the association for the initial assignment. Iteration of this process for all nodes on M completes the initialization of the LV segmentation. Note that the initialization is 1-to-1 mapping.

Figure 14a shows the segmentation result of both CA and LV : The parts corresponding to the branches of LCA are red-colored while those of RCA are blue-colored. Figure 14b and c shows the segmentations of LV in LCA side, and view outside and inside, respectively. Figure 14d-f shows the corresponding segmentations in RCA side, respectively. Recall that we model LV with a wall thickness: Be aware that LV inside is also segmented and thus the volume of LV region corresponding to a supplying CA piece can be measured.

Lemma 10. Given \mathcal{MAT} , $LV(V, E, F)$ can be segmented in $O(|V|\log|V|)$ time.

Proof: Computing $DT(P)$ takes $O(P\log P)$ time. With DT , an initial assignment can be found in $O(|P|)$ time. The segmentation can be done in $O(|C|)$ time, where $|C|$ denotes the number of cells in CDT.

Note that $|N| \ll |V|$ and $|C| = O(|V|)$. Therefore, the segmentation of LV takes $O(|V|\log|V|)$ time. \square

5.4. Summary of algorithm CardiacVis

The proposed algorithm is summarized in Algorithm 2. It first constructs the constrained Delaunay triangulation CDT^{CA} of CA (Step 1), which is used for the construction of an adjacency graph (Step 2). Step 3 extracts an adjacency tree by removing the cycles of the adjacency graph. The adjacency tree is transformed to the medial axis by removing outrageous nodes, shaving hair, and straightening bumpy nodes in Steps 4, 5, and 6, respectively. Step 7 computes CDT^{LV} for the segmentation of LV. Then, the algorithm segments LV and CA into subregions by assigning each tetrahedral cell in both CDT^{LV} and CDT^{CA} to each node in the medial axis in Steps 8 and 9, respectively.

Algorithm 2: Segmenting CA and LV

input : The mesh models of coronary artery CA and left ventricle LV
output: Medial axis MAT and the segmentation of CA and LV

- 1 Step 1) Construct the constrained Delaunay triangulation CDT^{CA} of CA;
 - 2 Step 2) Compute the adjacency graph G from CDT^{CA} ;
 - 3 Step 3) Extract the adjacency tree T^{BWD} from G (Algorithm 1);
 - 4 Step 4) Remove the outrageous nodes;
 - 5 Step 5) Shave hairs of T^{BWD} (Algorithm 3);
 - 6 Step 6) Straighten the bumpy nodes of T^{BWD} ;
 - 7 Step 7) Segment CA;
 - 8 Step 8) Compute constrained Delaunay triangulation CDT^{LV} of LV;
 - 9 Step 9) Segment LV;
-

Theorem 11. Given $CA(V_{CA}, E_{CA}, F_{CA})$ and $LV(V_{LV}, E_{LV}, F_{LV})$, the Medial-ABC algorithm takes $O(V_{CA}^2 + V_{LV} \log V_{LV})$.

6. Experiment and Discussion

This section presents experimental results using a set of 20 clinical cases of anonymous patients from Samsung Medical Center, Seoul, Korea. The cardiac CT images of LCA, RCA, and LV of each case were obtained using a dual source CT scanner, SOMATOM Definition Flash (Siemens Healthineers, Germany) (Sie, 2016), with a slice thickness of 0.6 mm and nonionic contrast medium (Iomeron) in DICOM (Digital Imaging and Communications in Medicine) format (Digital Imaging and Communications in Medicine, 2015). The mesh models were extracted from the CT images and stored in the STL format using the Vitrea Workstation program (Vitrea workstation, 2015). Refer to the accompanying STL files of the 20 test cases. The computational environment is as follows: Intel Core2 Duo E7500 2.93 Ghz, 4GB RAM, Windows 7, and Microsoft Visual C++.

6.1. Preview of the 20 test cases

The mesh models of the 20 test cases are referred to as (Samsung) Model- i , $i = 1, 2, \dots, 20$, where i corresponds to the size of left ventricle LV (i.e. #vertices): The more vertices an LV has, the

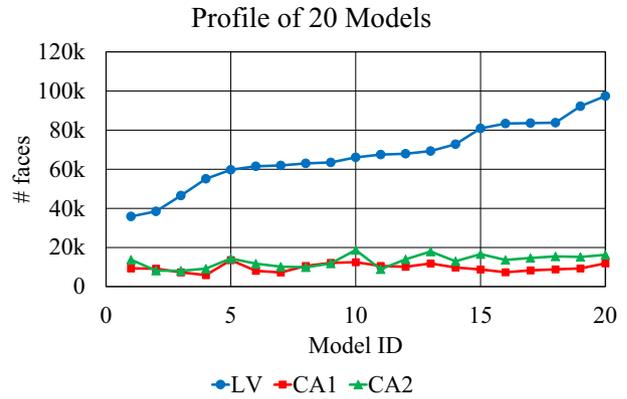


Figure 15: The statistics profile of 20 Samsung models.

higher its i is. This is because LV is the major muscle responsible for the most mechanical function of heart. Each model has three components: LV, CA1, and CA2, where CA1 or CA2 may be either the left or right coronary artery. Figure 15 shows the statistics profile of the 20 models.

Let $|V|$, $|E|$, and $|F|$ denote the number of vertices, edges, and faces in each model.

We first checked to confirm that the models are all 2-manifold and watertight. Next, we checked the Euler characteristic that is $|V| - |E| + |F| = 2(1 - g)$. The specific statistics of the 20 models are represented on Appendix D (Model-13 is the example we used throughout this paper). As coronary arteries are cycle-free in human body, it is expected to find $g = 0$ in both CA1 and CA2 for all models. However, the coronary arteries of some models have nonzero g s, implying the existence of topological handles. Appendix C shows some example of the topological handles from the test cases.

6.2. Center of circumsphere vs. center of mass

We used the center of the circumsphere of each CDT cell in the generation of the nodes of the adjacency graph, say G^{circum} . One may wonder what if we use the center of mass of each cell instead of the center of the circumsphere to get the adjacency graph, say G^{mass} . Obviously the topologies of both graphs are identical. It might seem that G^{mass} might be advantageous in that its nodes are guaranteed to lie inside of CA. However, it turns out that G^{mass} is not appropriate for the proposed algorithm but G^{circum} is as illustrated by the example below.

Figure 16a and c shows the shaved adjacency trees, say $T_{\text{circum}}^{\text{Shaved}}$ and $T_{\text{mass}}^{\text{Shaved}}$, corresponding to G^{circum} and G^{mass} , respectively. Figure 16b and d shows the zoom-ups of an arbitrary segment. Observe that the quality of $T_{\text{mass}}^{\text{Shaved}}$ is significantly inferior to that of $T_{\text{circum}}^{\text{Shaved}}$ for two important view points. First, the path of $T_{\text{mass}}^{\text{Shaved}}$ is very much wiggling. Second and more critically, the path of $T_{\text{mass}}^{\text{Shaved}}$ is quite significantly off-the-center of the artery to the mesh boundary because CDT frequently contains many flat-shaped cells that lie near boundary.

6.3. Properties of the Medial-ABC method

Figure 17a shows the size of the adjacency graph and adjacency tree of CA: Different colors denote the entity sizes after different steps are applied. Note that the data size decreases as the algorithm proceeds its steps. The adjacency tree extraction and shaving steps significantly reduce the number of nodes: The out-

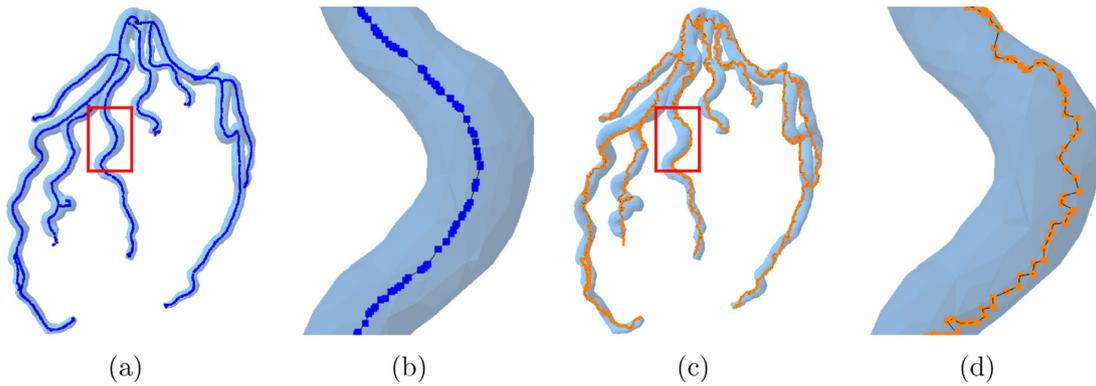


Figure 16: Generated MAT from two different adjacency graphs. (a) MAT when center of circumsphere is used (proposed method). (b) Close-up of (a) in red box. (c) MAT when center of mass is used. (d) Close-up of (c) in red box.

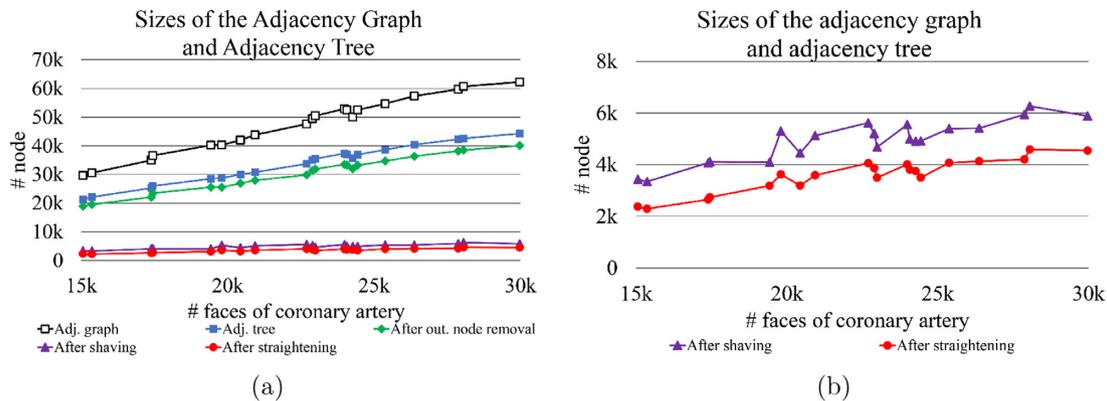


Figure 17: Size of the adjacency graph and adjacency tree: The data size decreases as the algorithm proceeds in its steps for computing the medial axis. (a) Different colors denote the entity size after different steps of the algorithm are applied and (b) only two curves corresponding to both shaving and straightening are shown.

rageous node removing and the straightening steps do not reduce much. Figure 17b shows two curves corresponding to the shaving and straightening steps.

Figure 18 shows the computation time of both the medial axis and the entire segmentation for the test data set of the 20 clinical cases. We group the steps of Algorithm 2 into three phases: Phase I (extraction of adjacency tree) consisting of Steps 1, 2, and 3; Phase II (transformation of adjacency tree to medial axis) consisting of Steps 4, 5, and 6; and Phase III (segmentation of ventricles and coronary arteries) consisting of Steps 7, 8, and 9. Figure 18a decomposes the computation time for the medial axis into three parts: the time for loading each CA model file, the time for Phase I, and the time for Phase II. Note that the total time shows a quadratic increase with respect to the model size because the most expensive operation is Step 3 with the time complexity $O(|N|^2)$ with respect to $G(N, L)$ (see Section 3.2) and $|N|$ linearly increases regarding the model size. For the clarity of the other computation times, the times for Phase I and II are shown in Fig. 18b and c, respectively. The time for loading the CA triangular mesh model is excluded. Among the times for Phase I, the time for adjacency tree extraction is mostly dominant while times for both CDT and the adjacency graph are relatively negligible. For the steps of Phase II, the outrageous node removing takes more time than other steps. The bumpy-node-straightening step is relatively negligible. Figure 18d shows the times for the LV model file loading and Phase III, which consists of the computation of CDT^{LV} , CA segmentation, and LV segmentation. The time for CDT^{CA} is not shown because that is already

included in Phase I. The time for LV segmentation shows a super linear increase with model size because its time complexity is $O(|V|\log|V|)$ with respect to $LV(V, E, F)$ (refer lemma 10).

7. Developed Program: CardiacVis

We have developed the CardiacVis program, which implemented the proposed algorithm on Microsoft Windows. Figure 19 shows the segmentation of CA and LV in the main graphics window. After segmenting LV and CA using MAT , CardiacVis visualizes the recognized branches of CA in the main pane with the synchronized colors for the corresponding LV and CA segments. The right pane displays the tree hierarchy of CA branches and the lower pane shows the parent-child relationship between CA branches with a color encoding. The lower pane also shows the mass properties of each segmented subregion of LV and its supplying CA branch. The mass properties include the volume/surface area of the segmented LV subregion, the length, thickness, surface area, and volume of CA branch, of which the importance for medical diagnosis and treatment to the cardiac function was verified (Frangi et al., 2001; Kurata et al., 2015; Sumitsuji et al., 2016). Hence, the CardiacVis program, which implemented the proposed algorithm, will be useful for the assessment of the severity of heart attack by quantifying the volume and area of the myocardium at risk.

Figure 20 shows an application using the segmentation of LV. Suppose that we pick a point, as marked by the yellow arrow, which actually corresponds to a node of MAT . Consider that CA

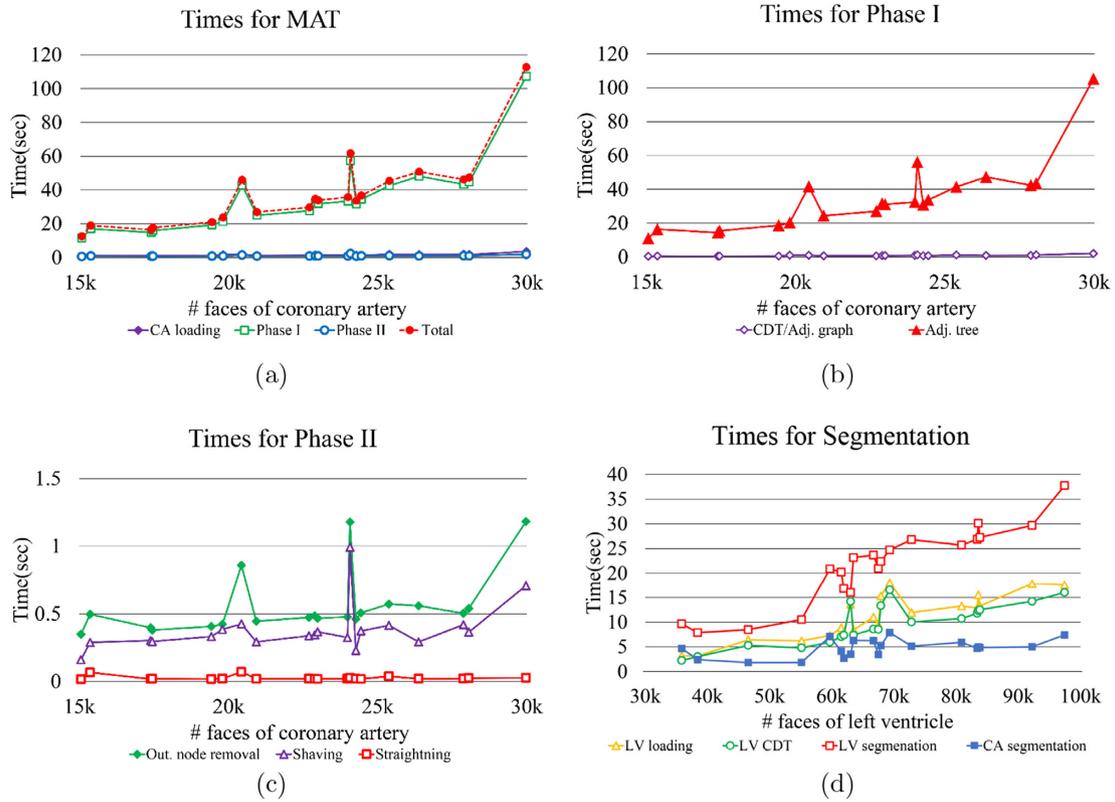


Figure 18: Computation time for test data set. (a) Decomposition of the medial axis computation time into CA model file loading, Phase I, and Phase II, (b) decomposition of times for Phase I into CDT/adjacency graph and adjacency tree, (c) decomposition of times for Phase II into outrageous node removing, hair shaving, and bumpy node straightening, and (d) computation time for LV model file loading, Phase III, which is decomposed into CDT^{LV} of LV, LV segmentation, and CA segmentation.

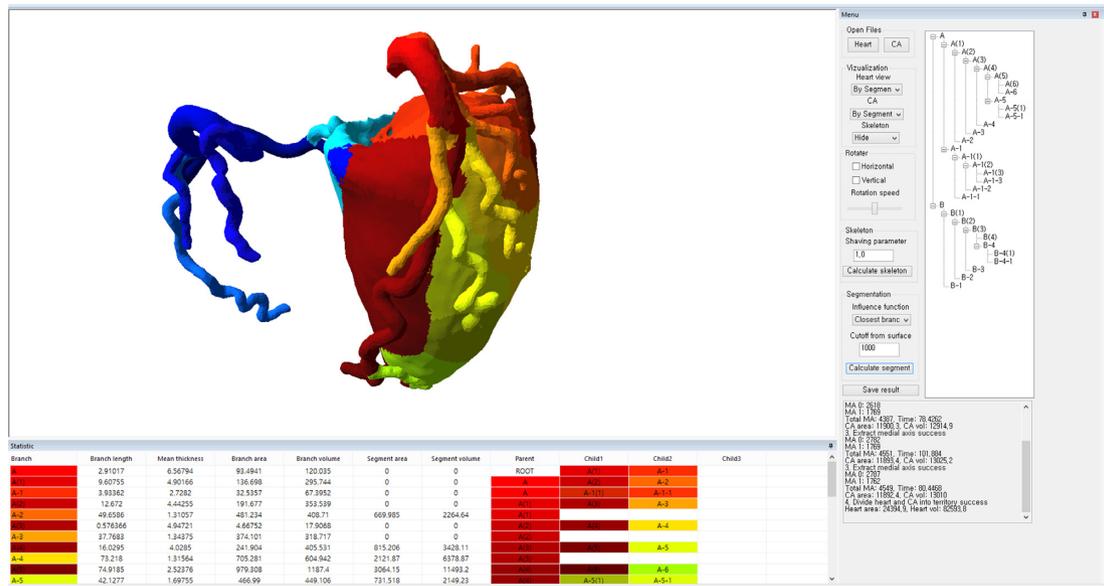


Figure 19: The CardiacVis program to segment the mesh models LV and CA of the left ventricle and coronary artery, respectively. After segmenting LV and CA using MAT, the CardiacVis program displays the recognized CA branches and the corresponding segmented LV subregions with the colors synchronized in the main pane. The right pane shows the tree hierarchy of CA branches. The lower pane shows the mass properties of each segmented subregion of LV and its supplying CA branch. The mass properties include the volume/surface area of the segmented LV subregion, the length, thickness, surface area, and volume of the CA branch.

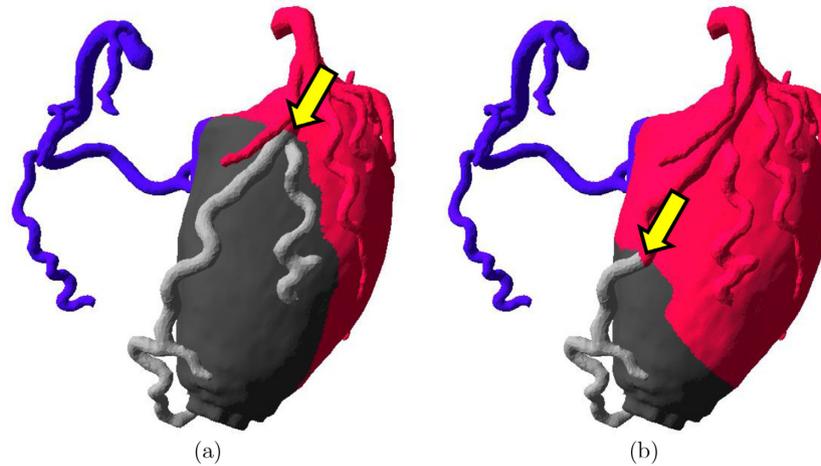


Figure 20: Linkage between the mesh model LV of left ventricle and the mesh model CA of a supplying coronary artery. Given the segmentation of LV and CA, myocardium at risk can be precisely localized and quantified by designating the location of obstruction (yellow arrows). (a) Obstruction in proximal CA results in larger amounts of myocardium at risk compared to (b) obstruction in distal CA. The extent and border of myocardium at risk can also be clearly identified. The obstructive CA pieces are shown in light gray and the subtended LV pieces in dark green.

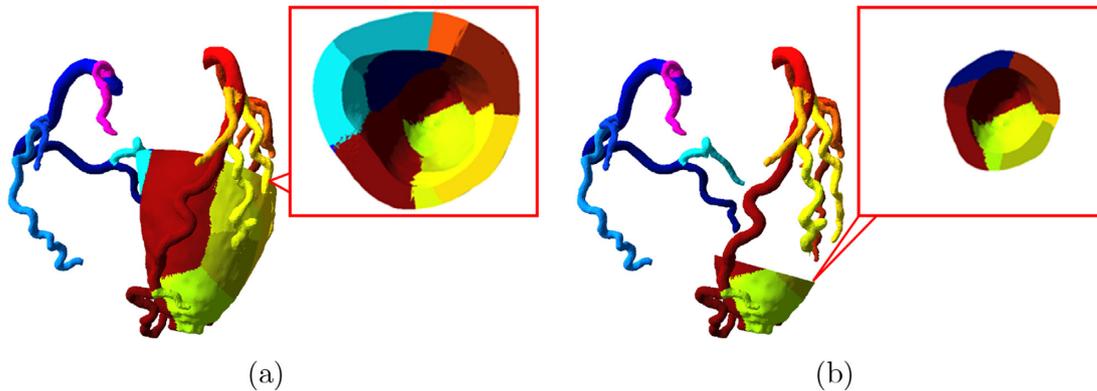


Figure 21: Section view of a left ventricle. After the upper lump of the left ventricle above a trimming plane is removed, the remaining part of the left ventricle and the entire coronary arteries are shown in (a). The trimming plane is created through a user interaction with *CardiacVis* via screen (The trimming plane is not shown). The figure in the red box of (a) is the ventricle from a different view: Be aware that the muscle in the ventricle wall is also properly assigned to a corresponding CA branch. (b) similarly shows a section view for a different trimming plane.

is obstructed at the picked point. Figure 20a shows (i) the subset of the coronary artery from the pick point down to the leaves (shown in light gray), and (ii) the subset of LV corresponding to CA subset (shown in dark gray). Figure 20b shows the case that the picking point is located further down to a leaf. Observe that the corresponding LV region shrinks.

Figure 21 shows a section view of the left ventricle and the entire coronary arteries. Figure 21a shows the remaining part of the left ventricle after the upper ventricular lump above a trimming plane is removed where the trimming plane (not shown in the figure) is created through a user interaction with *CardiacVis* via the screen. The figure in the red box of Fig. 21a is the ventricle from a different view. Be aware that the muscle in the ventricle wall is also properly assigned to a corresponding CA branch. Figure 21b similarly shows a section view for a different trimming plane. Thus, we can investigate the morphometry of myocardium such as the wall thickness from the section view and can easily compute the thickness if necessary. Note that the wall thickness of the left ventricle is an important measure for analyzing cardiac function and diagnosing cardiovascular disease (Sasayama, Franklin, Ross, Kemper, & McKown, 1976; Gaasch, 1979; Olivetto et al., 2003).

Thus, the proposed mesh model-based approach can facilitate various clinical studies where model quantification is an important measure (Frangi et al., 2001; Prakash & Ethier, 2001; Saito et al., 2005; Kurata et al., 2015; Sumitsuji et al., 2016). Furthermore, the proposed research could be exploited for applications related to model optimization. For example, one of the promising therapies for cardiac disease is to transplant stem cells into either the myocardium at the site of injury or the supplying CA branch (Segers & Lee, 2008; Shafiq et al., 2016). One important issue for this approach is to optimize the delivery of stem cells to the appropriate site so that cardiac regeneration is maximized (Oettgen, 2006). In this case, the segmentation result of this study would be more importantly used for the delivery optimization.

8. Conclusion

This study presents an algorithm and its implementation to segment regional myocardium at risk subtended by any potentially obstructed CA based on the geometric models of a triangular mesh for the CA and LV obtained from an individual cardiac CT

image. The key idea of the Medial-ABC algorithm is (i) computation of MAT of CA and (ii) segmentation of CA and LV into a set of regions where each corresponds to a node of the medial axis. The MAT is transformed from an adjacency tree, which is extracted by removing cycles of an adjacency graph. The adjacency graph is constructed from the constrained Delaunay triangulation of the triangular mesh model of the coronary artery. The algorithmic accuracy and efficiency are theoretically asserted and experimentally verified.

Obstruction of the CA results in acute myocardial infarction. Hence, quantification of the regional amount of myocardium subtended by the obstructed CA is of critical value in clinical medicine. However, conventional methods such as the 17-piece model are inaccurate and frequently disagree with clinical practice. The proposed algorithm provides a robust mathematical linkage between myocardium at risk and supplying CA so that ischemic myocardial region can be accurately identified, and both the extent and severity of myocardial ischemia can be quantified effectively and efficiently. Furthermore, the computed result of segmented CA and LV can be more importantly used for building optimization models of cardiac systems for various applications. We believe that the algorithm and developed CardiacVis program will be an invaluable tool for patient-specific risk predictions and the treatment of obstructed CA disease in clinical medicine.

The clinical benefit of the proposed research can be summarized into the following two aspects:

- Medial-ABC enables mathematical interpretation of coronary artery disease, which would enhance physician's understanding of coronary artery disease and support machine-based automatic diagnosis and decision of treatment.
- Medial-ABC enables myocardial tessellation, which is required for computational modeling of coronary circulation. Now, several commercial products including FFR-CT have been developed and are being used in clinical practice.

Even though we do not mention in this paper, the accuracy evaluation of the proposed research is important. Hence, the concept is not so simple; we reasoned that the whole study process needs to be separated into stepwise processes. As a proof of concept, here we presented our methodology and representative cases. Validation of accuracy needs large amount of clinical data, which would require extensive workload and data acquisition cost. Considering the amount of data to be presented with validation, it might be appropriate to present it in a separate paper.

Acknowledgements

J.C., J.R., and D.-S.K. were supported by the National Research Foundation of Korea grant funded by MSIP (2017R1A3B1023591, 2016K1A4A3914691), Korea. J.-H.C. was supported by grants from the Samsung Medical Center basic research (GL1B33211, CRP1500053), the Korean Circulation Society Research Fund (201301-01), the Korean Society of Interventional Cardiology (2014-1), and the National Research Foundation of Korea (2017R1A2B310918). The authors would like to thank research assistants Seon-A Jeong and So-Hyeon Park for their efforts to prepare clinical data.

Conflict of interest statement

None declared.

REFERENCES

- Amenta, N., Bern, M., & Kamvysselis, M. (1998). A new Voronoi-based surface reconstruction algorithm. In *Proceedings of the SIGGRAPH '98* (pp. 415–421).
- Amenta, N., Choi, S., & Kolluri, R. K. (2001). The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications*, 19, 127–153.
- Attali, D., & Montanvert, A. (1997). Computing and simplifying 2D and 3D continuous skeletons. *Computer Vision and Image Understanding*, 67(3), 261–273.
- Aurenhammer, F. (1991). Voronoi diagrams – A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3), 345–405.
- Bae, Y. G., Hwang, S. T., Han, H., Kim, S. M., Kim, H.-Y., Park, I., Lee, J. M., Moon, Y.-J., & Choi, J.-H. (2018). Non-invasive coronary physiology based on computational analysis of intracoronary transmural attenuation gradient. *Scientific Reports*, 8(1), 4692.
- beng Ho, S., & Dyer, C. R. (1986). Shape smoothing using medial axis properties. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(4), 512–520.
- Blum, H. (1967). A transformation for extracting new descriptors of shaped. In W. Wathen-Dunn (Ed.), *Models for the perception of speech and visual form* (pp. 362–380), Cambridge, MA: MIT Press.
- Blum, H., & Nagel, R. N. (1978). Shape description using weighted symmetric axis features. *Pattern Recognition*, 10, 167–180.
- Borkin, M. A., Gajos, K. Z., Peters, A., Mitsouras, D., Melchionna, S., Rybicki, F. J., Feldman, C. L., & Pfister, H. (2011). Evaluation of artery visualizations for heart disease diagnosis. *IEEE Transactions on Visualization and Computer Graphics*, 17(12), 2479–2488.
- Brandt, J. (1994). Convergence and continuity criteria for discrete approximation of the continuous planar skeleton. *CVGIP: Image Understanding*, 59(1), 116–124.
- Bruyne, B. D., Fearon, W. F., Pijls, N. H., Barbato, E., Tonino, P., Piroth, Z., Jagic, N., Möbius-Winckler, S., Rioufol, G., Witt, N., Kala, P., MacCarthy, P., Engström, T., Oldroyd, K., Mavromatis, K., Manoharan, G., Verlee, P., Frobert, O., Curzen, N., Johnson, J. B., Limacher, A., Nüesch, E., & Jüni, P. (2014). Fractional flow reserve-guided PCI for stable coronary artery disease. *The New England Journal of Medicine*, 371, 1208–1217.
- Bruyne, B. D., Pijls, N. H., Kalesan, B., Barbato, E., Tonino, P. A., Piroth, Z., Jagic, N., Möbius-Winkler, S., Rioufol, G., Witt, N., Kala, P., MacCarthy, P., Engström, T., Oldroyd, K. G., Mavromatis, K., Manoharan, G., Verlee, P., Frobert, O., Curzen, N., Johnson, J. B., Jüni, P., & Fearon, W. F. (2012). Fractional flow reserve-guided PCI versus medical therapy in stable coronary disease. *The New England Journal of Medicine*, 367, 991–1001.
- Cai, Y., Ye, X., Chui, C., & Anderson, J. H. (2003). Constructive algorithms of vascular network modeling for training of minimally invasive catheterization procedure. *Advances in Engineering Software*, 34(7), 439–450.
- Cerqueira, M. D., Weissman, N. J., Dilsizian, V., Jacobs, A. K., Sanjiv, Kaul M., Laskey, W. K., Pennell, D. J., Rumberger, J. A., Ryan, T., & Verani, M. S. (2002). Standardized myocardial segmentation and nomenclature for tomographic imaging of the heart. *Circulation*, 105, 539–542.
- Chew, L. P. (1989). Constrained Delaunay triangulation. *Algorithmica*, 4(1–4), 97–108.
- Choi, S. W., & Seidel, H.-P. (2004). Linear one-sided stability of mat for weakly injective 3D domain. *Computer-Aided Design*, 36(2), 95–109.

- Cignoni, P., Callieri, M., Corsini, M., Dellepiane, M., Ganovelli, F., & Ranzuglia, G. (2008). MeshLab: An open-source mesh processing tool. In *Eurographics Italian Chapter Conference*, Salerno, Italy.
- Cornea, N. D., Silver, D., & Min, P. (2007). Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics*, 13(3), 530–548.
- Culver, T., Keyser, J., & Manocha, D. (2004). Exact computation of the medial axis of a polyhedron. *Computer Aided Geometric Design*, 21(1), 65–98.
- Dawson, T. (2005). Modeling of vascular networks. *Journal of Experimental Biology*, 208, 1687–1694.
- Debarba, H. G., Zanchet, D. J., Fracaro, D., Maciel, A., & Kalil, A. N. (2010). Efficient liver surgery planning in 3D based on functional segment classification and volumetric information. In *Proceedings of the 32nd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*.
- de Putter, S., van de Vosse, F. N., Gerritsen, F. A., Laffargue, F., & Breeuwer, M. (2006). Computational mesh generation for vascular structures with deformable surfaces. *International Journal of Computer Assisted Radiology and Surgery*, 1(1), 39–49.
- Dey, T. K., & Zhao, W. (2004). Approximate medial axis as a Voronoi subcomplex. *Computer-Aided Design*, 36, 195–202.
- Digital Imaging and Communications in Medicine (2015). <https://www.dicomstandard.org/>. Accessed 14 July 2020.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.
- Frangi, A. F., Niessen, W. J., & Viergever, M. A. (2001). Three-dimensional modeling for functional analysis of cardiac images: A review. *IEEE Transactions on Medical Imaging*, 20(1), 2–25.
- Fredman, M. L., & Tarjan, R. (1984). Fibonacci heaps and their uses in improved network optimization algorithms. In *Proceedings of the 25th Annual Symposium on Foundations of Computer Science* (pp. 338–346).
- Gaasch, W. H. (1979). Left ventricular radius to wall thickness ratio. *The American journal of cardiology*, 43(6), 1189–1194.
- Glaber, S., Lawonn, K., Hoffmann, T., Skalej, M., & Preim, B. (2014). Combined visualization of wall thickness and wall shear stress for the evaluation of aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 20(12), 2506–2515.
- Gong, W., & Bertrand, G. (1990). A simple parallel 3D thinning algorithm. In *Proceedings of the IEEE Pattern Recognition* (pp. 188–190).
- Han, H., Bae, Y. G., Hwang, S. T., Kim, H.-Y., Park, I., Kim, S.-M., Choe, Y., Moon, Y.-J., & Choi, J.-H. (2018). Computationally simulated fractional flow reserve from coronary computed tomography angiography based on fractional myocardial mass. *The International Journal of Cardiovascular Imaging*, 35(1), 185–193.
- Holleman, C., & Kavraki, L. E. (2000). A framework for using the workspace medial axis in PRM planners. In *Proceedings of the IEEE International Conference on Robotics and Automation* (Vol. 2, 1408–1413).
- Imai, Y., Hiraoka, H., & Kawaharada, H. (2014). Quadrilateral mesh fitting that preserves sharp features based on multi-normals for Laplacian energy. *Journal of Computational Design and Engineering*, 1(2), 88–95.
- Jalba, A. C., Kustra, J., & Telea, A. C. (2013). Surface and curve skeletonization of large 3D models on the GPU. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6), 1495–1508.
- Javadi, M. S., Lautamäki, R., Merrill, J., Voicu, C., Epley, W., McBride, G., & Bengel, F. M. (2010). Definition of vascular territories on myocardial perfusion images by integration with true coronary anatomy: A hybrid pet/ct analysis. *Journal of Nuclear Medicine*, 51(2), 198–203.
- Kim, D.-S. (1998). Polygon offsetting using a Voronoi diagram and two stacks. *Computer-Aided Design*, 30(14), 1069–1076.
- Kim, J.-K., Cho, Y., Kim, D., & Kim, D.-S. (2014). Voronoi diagrams, quasi-triangulations, and beta-complexes for disks in \mathbb{R}^2 : The theory and implementation in BetaConcept. *Journal of Computational Design and Engineering*, 1(2), 79–87.
- Kim, H. Y., Doh, J.-H., Lim, H.-S., Nam, C.-W., Shin, E.-S., Koo, B.-K., Lee, J. M., Park, T. K., Yang, J. H., Song, Y. B., Hahn, J.-Y., Choi, S. H., Gwon, H.-C., Lee, S.-H., Kim, S. M., Choe, Y., & Choi, J.-H. (2017). Identification of coronary artery side branch supplying myocardial mass that may benefit from revascularization. *JACC: Cardiovascular Interventions*, 10(6), 571–581.
- Kim, D.-S., Hwang, I.-K., & Park, B.-J. (1995). Representing the Voronoi diagram of a simple polygon using rational quadratic Bézier curves. *Computer-Aided Design*, 27(8), 605–614.
- Kim, D.-S., Kim, D., Cho, Y., & Sugihara, K. (2006). Quasi-triangulation and interworld data structure in three dimensions. *Computer-Aided Design*, 38(7), 808–819.
- Kim, H. Y., Lim, H.-S., Doh, J.-H., Nam, C.-W., Shin, E.-S., Koo, B.-K., Yoon, M.-H., Tahk, S.-J., Kang, D. K., Song, Y. B., Hahn, J.-Y., Choi, S. H., Gwon, H.-C., Lee, S.-H., Kim, E.-K., Kim, S. M., Choe, Y., & Choi, J.-H. (2016). Physiological severity of coronary artery stenosis depends on the amount of myocardial mass subtended by the coronary artery. *JACC: Cardiovascular Interventions*, 9(15), 1548–1560.
- Kim, S., Sohn, D., & Im, S. (2019). Construction of polyhedral finite element meshes based upon marching cube algorithm. *Advances in Engineering Software*, 128, 98–112.
- Kirkpatrick, D. G. (1979). Efficient computation of continuous skeletons. In *Proceedings of the 14th Annual IEEE Symposium on Foundations of Computer Science* (pp. 18–27).
- Kurata, A., Kono, A., Sakamoto, T., Kido, T., Mochizuki, T., Hishino, H., Abe, M., Coenen, A., Saru-Chelu, R. G., de Feyter P., Krestin, G. P., & Nieman, K. (2015). Quantification of the myocardial area at risk using coronary CT angiography and Voronoi algorithm-based myocardial segmentation. *European Radiology*, 25(1), 49–57.
- Lee, D. T. (1982). Medial axis transformation of a planar shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4, 363–369.
- Lee, M., Fang, Q., Cho, Y., Ryu, J., Liu, L., & Kim, D.-S. (2018). Support-free hollowing for 3D printing via Voronoi diagram of ellipses. *Computer-Aided Design*, 101, 23–36.
- Leopold, J. A., & Loscalzo, J. (2018). Emerging role of precision medicine in cardiovascular disease. *Circulation Research*, 122(9), 1302–1315.
- Lesage, D., Angelini, E. D., Bloch, I., & Funka-Lea, G. (2009). A review of 3D vessel lumen segmentation techniques: Models, features and extraction schemes. *Medical Image Analysis*, 13(6), 819–845.
- Linardakis, L., & Chrisochoides, N. (2008). Algorithm 870: A static geometric medial axis domain decomposition in 2D Euclidean space. *ACM Transactions on Mathematical Software*, 34(1), 4.
- Liu, L., Chambers, E. W., Letscher, D., & Ju, T. (2010). A simple and robust thinning algorithm on cell complexes. *Computer Graphics*, 29(7), 2253–2260.
- Lorenz, C., Renisch, S., Schlathölter, T., & Bülow, T. (2003). Simultaneous segmentation and tree reconstruction of the coronary arteries in MSCT images. In *Proceedings of SPIE - The*

- International Society for Optical Engineering (Vol. 5031, pp. 167–177).
- Lorenz, C., & von Berg, J. (2006). A comprehensive shape model of the heart. *Medical Image Analysis*, 10, 657–670.
- Mangion, K., Gao, H., Husmeier, D., Luo, X., & Berry, C. (2018). Advances in computational modelling for personalised medicine after myocardial infarction. *Heart*, 104, 550–557.
- Meyer-Spradow, J., Stegger, L., Döring, C., Ropinski, T., & Hinrichs, K. (2008). Glyph-based SPECT visualization for the diagnosis of coronary artery disease. *IEEE Transactions on Visualization and Computer Graphics*, 14(6), 1632–1639.
- Mitchell, S. C., Lelieveldt, B. P. F., van der Geest, R. J., Bosch, H. G., Reiber, J. H. C., & Sonka, M. (2001). Multistage hybrid active appearance model matching: Segmentation of left and right ventricles in cardiac MR images. *IEEE Transactions on Medical Imaging*, 20(5), 415–423.
- Moon, S., & Ko, K. (2018). A point projection approach for improving the accuracy of the multilevel B-spline approximation. *Journal of Computational Design and Engineering*, 5(2), 173–179.
- Nackman, L. R., & Pizer, S. M. (1985). Three-dimensional shape description using the symmetric axis transform I: Theory. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(2), 187–202.
- Oeltze, S., Doleisch, H., Hauser, H., Muigg, P., & Preim, B. (2007). Interactive visual analysis of perfusion data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1392–1399.
- Oettgen, P. (2006). Cardiac stem cell therapy need for optimization of efficacy and safety monitoring. *Circulation*, 114(4), 339–358.
- Oh, S. (2019). A new triangular mesh repairing method using a mesh distortion energy minimization-based mesh flattening method. *Advances in Engineering Software*, 131, 48–59.
- Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. (1999). *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. (2nd ed.). Chichester: John Wiley & Sons.
- Olivotto, I., Gistri, R., Petrone, P., Pedemonte, E., Vargiu, D., & Cecchi, F. (2003). Maximum left ventricular thickness and risk of sudden death in patients with hypertrophic cardiomyopathy. *Journal of the American College of Cardiology*, 41(2), 315–321.
- Ortiz-Pérez, J. T., Rodríguez, J., Meyers, S. N., Lee, D. C., Davidson, C., & Wu, E. (2008). Correspondence between the 17-segment model and coronary arterial anatomy using contrast-enhanced cardiac magnetic resonance imaging. *JACC: Cardiovascular Imaging*, 1(3), 282–293.
- Paragios, N. (2003). A level set approach for shape-driven segmentation and tracking of the left ventricle. *IEEE Transactions on Medical Imaging*, 22(6), 773–776.
- Park, J. M., Lee, B. C., Chae, S. W., & Kwon, K. Y. (2019). Surface reconstruction from FE mesh model. *Journal of Computational Design and Engineering*, 6(2), 197–208.
- Pizer, S. M., Oliver, W. R., & Bloomberg, S. H. (1987). Hierarchical shape description via the multiresolution symmetric axis transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4), 505–511.
- Prakash, S., & Ethier, C. R. (2001). Requirements for mesh resolution in 3D computational hemodynamics. *Journal of Biomechanical Engineering*, 123, 134–144.
- Razavi, M. S., Shirani, E., & Kassab, G. S. (2018). Scaling laws of flow rate, vessel blood volume, lengths, and transit times with number of capillaries. *Frontiers in Physiology*, 9, 581.
- Saito, S., Miura, J. Y. K., Nakao, N., Nagao, T., Sugimoto, T., Hirano, T., Kuroda, N., Iimuro, Y., & Fujimoto, J. (2005). A novel 3D hepatectomy simulation based on liver circulation: Application to liver resection and transplantation. *Hepatology*, 41(6), 1297–1304.
- Sasayama, S., Franklin, D., Ross, J. Jr., Kemper, W. S., & McKown, D. (1976). Dynamic changes in left ventricular wall thickness and their use in analyzing cardiac function in the conscious dog: A study based on a modified ultrasonic technique. *The American Journal of Cardiology*, 38(7), 870–879.
- Segers, V. F. M., & Lee, R. T. (2008). Stem-cell therapy for cardiac disease. *Nature*, 451, 937–942.
- Shafiq, M., Jung, Y., & Kim, S. H. (2016). Insight on stem cell preconditioning and instructive biomaterials to enhance cell adhesion, retention, and engraftment for tissue repair. *Biomaterials*, 90, 85–115.
- Shaked, D., & Brucksteiny, A. M. (1998). Pruning medial axes. *Computer Vision and Image Understanding*, 69(2), 156–169.
- Shewchuk, J. R. (2003). Updating and constructing constrained Delaunay and constrained regular triangulations by flips. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, (pp. 181–190).
- Si, H. (2008). Three-dimensional boundary conforming Delaunay mesh generation. Ph.D. thesis, TU Berlin.
- Si, H. (2013). Tetgen: A quality tetrahedral mesh generator and 3D Delaunay triangulator, Technical Report 13, WIAS.
- Si, H. (2015). Tetgen, a Delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software*, 41(2), 1–36.
- Siemens Healthineers (2016). <https://www.siemens-healthineers.com/>. Accessed 14 July 2020.
- Si, H., & Gartner, K. (2011). 3D boundary recovery by constrained Delaunay tetrahedralization. *International Journal for Numerical Methods in Engineering*, 85, 1341–1364.
- Si, H., & Shewchuk, J. R. (2014). Incrementally constructing and updating constrained Delaunay tetrahedralizations with finite-precision coordinates. *Engineering with Computers*, 30, 253–269.
- Sugihara, K. (1992). A simple method for avoiding numerical errors and degeneracy in Voronoi diagram construction. *IEICE Transactions. Fundamentals*, E75-A, 468–477.
- Sugihara, K., & Iri, M. (1989). A solid modelling system free from topological inconsistency. *Journal of Information Processing*, 12(4), 380–393.
- Sugihara, K., & Iri, M. (1992). Construction of the Voronoi diagram for “one million” generators in single-precision arithmetic. *Proceedings of the IEEE*, 80(9), 1471–1484.
- Sugihara, K., & Iri, M. (1994). A robust topology-oriented incremental algorithm for Voronoi diagrams. *International Journal of Computational Geometry & Applications*, 4(2), 179–228.
- Sumitsuji, S., Ide, S., Siegrist, P. T., Salah, Y., Yokoi, K., Yoshida, M., Awata, M., Yamasaki, K., Tachibana, K., Kaneda, H., Nanto, S., & Sakata, Y. (2016). Reproducibility and clinical potential of myocardial mass at risk calculated by a novel software utilizing cardiac computed tomography information. *Cardiovascular Intervention and Therapeutics*, 31, 218–225.
- Suri, J. S. (2000). Computer vision, pattern recognition and image processing in left ventricle segmentation: The last 50 years. *Pattern Analysis & Applications*, 3(3), 209–242.
- Tam, T., & Armstrong, C. (1991). 2D finite element mesh generation by medial axis subdivision. *Advances in Engineering Software and Workstations*, 13(5–6), 313–324.
- Termeer, M., Bescós, J. O., & Breeuwer, M. (2007). CoViCAD: Comprehensive visualization of coronary artery disease. *IEEE Transactions on Visualization and Computer Graphics*, 13(6), 1632–1639.

- Termeer, M., Bescós, J. O., Breeuwer, M., Vilanova, A., Gerritsen, F., Gröller, M. E., & Nagel, E. (2008). Patient-specific mappings between myocardial and coronary anatomy. In *Scientific visualization: Advanced concepts*, (pp. 196–209). Germany: Dagstuhl Publishing.
- Termeer, M., Bescós, J. O., Breeuwer, M., Vilanova, A., Gerritsen, F., Gröller, M., & Nagel, E. (2010). Patient-specific mappings between myocardial and coronary anatomy). In *Scientific visualization: Advanced concepts* (Vol. 1). Germany: Dagstuhl Publishing.
- Updegrove, A., Wilson, N. M., & Shadden, S. C. (2016). Boolean and smoothing of discrete polygonal surfaces. *Advances in Engineering Software*, 95, 16–27.
- Vitrea workstation (2015). <https://www.vitalimages.com/vitrea/>. Accessed 14 July 2020.
- Wade, L., & Parent, R. E. (2002). Automated generation of control skeletons for use in animation. *The Visual Computer*, 18(2), 97–110.
- Wei, X., Qiu, S., Zhu, L., Feng, R., Tian, Y., Xi, J., & Zheng, Y. (2018). Toward support-free 3D printing: A skeletal approach for partitioning models. *IEEE Transactions on Visualization and Computer Graphics*, 24(10), 2799–2812.
- West, G. B., Brown, J. H., & Enquist, B. J. (1997). A general model for the origin of allometric scaling laws in biology. *Science*, 276(5309), 122–126.
- Wischgoll, T., Choy, J. S., Ritman, E. L., & Kassab, G. S. (2008). Validation of image-based method for extraction of coronary morphometry. *Annals of Biomedical Engineering*, 36(3), 356–368.
- World Health Organization (2012). World health statistics 2012 (world health statistics annual). Technical report. World Health Organization.
- Xiong, G., Sun, P., Zhou, H., Ha, S., Hartaigh, B. O., Truong, Q.

A., & Min, J. K. (2017). Comprehensive modeling and visualization of cardiac anatomy and physiology from CT imaging and computer simulations. *IEEE Transactions on Visualization and Computer Graphics*, 23(2), 1014–1028.

Xu, J., Zhou, M., Wu, Z., Shui, W., & Ali, S. (2015). Robust surface segmentation and edge feature lines extraction from fractured fragments of relics. *Journal of Computational Design and Engineering*, 2(2), 79–87.

Yap, C.-K. (1997). Towards exact geometric computation. *Computational Geometry: Theory and Applications*, 7(1–2), 3–23.

Appendix A: Shaving Algorithm

The shaving process is done as follows. Suppose that n_i is ordered by the non-increasing length of π_i (i.e. π_1 is the longest). Let $T_0 \subseteq T^{\text{BWD}}$ be the tree consisting of the root node of T^{BWD} . We grow T_0 by concatenating every path π_i until the last one as follows: $T_i = \pi_i \oplus T_{i-1}$ (refer to Definition 5). Let $\lambda_{ij} = \lambda(\pi_j, T_i)$ be the length of the subset of π_j from a leaf node n_j to the nearest node of T_i . Then, we define the distance Dist_i between the subtree T_i and T^{BWD} as follows.

Definition 8. (Distance between trees) Consider the subtree $T_i \subseteq T^{\text{BWD}}$.

$$\text{Dist}_i = \text{Dist}(T_i, T^{\text{BWD}}) = \sum \lambda_{ij}, n_j \in N^{\text{Leaf}}, \quad (\text{A.1})$$

where N^{Leaf} denotes the set of leaf nodes in T^{BWD} .

Lemma 12. $\text{Dist}_i = 0$ if, and only if, $T_i \equiv T^{\text{BWD}}$; else, $\text{Dist}_i > 0$.

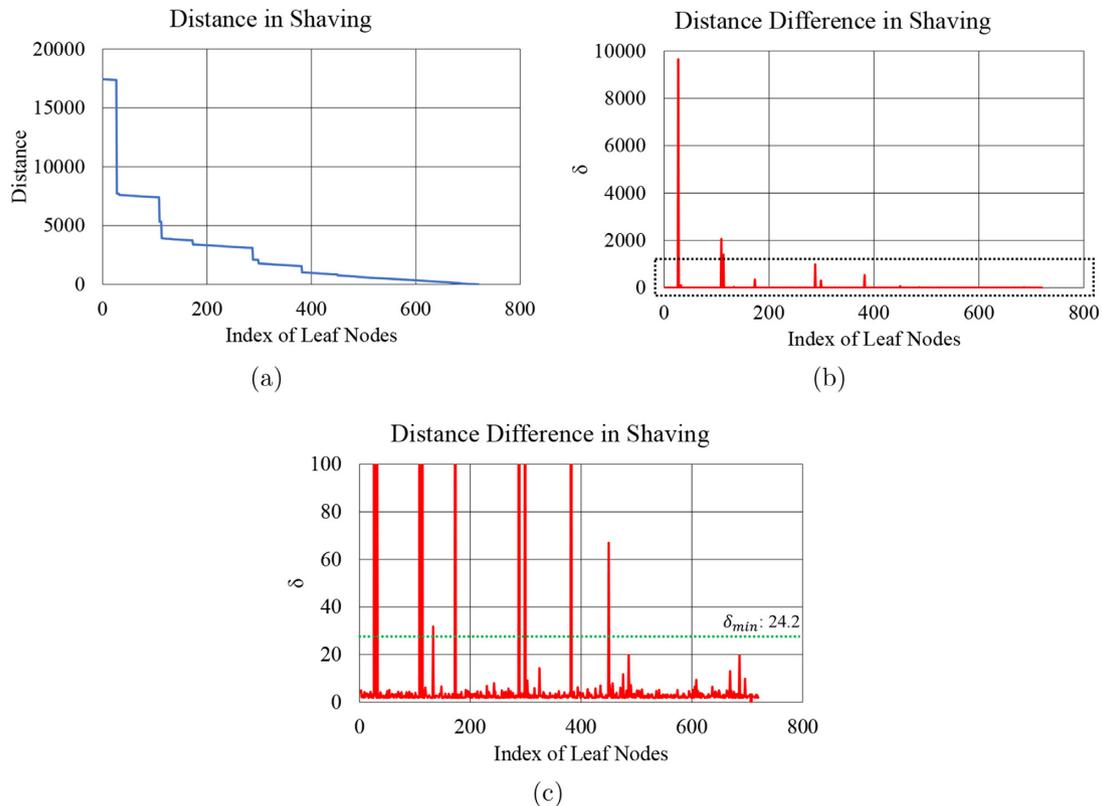


Figure A1: The profile of Dist_i and δ_i for shaving hairy nodes in LCA. (a) Dist_i . (b) δ_i . (c) Close-up with a threshold value δ_{\min} .

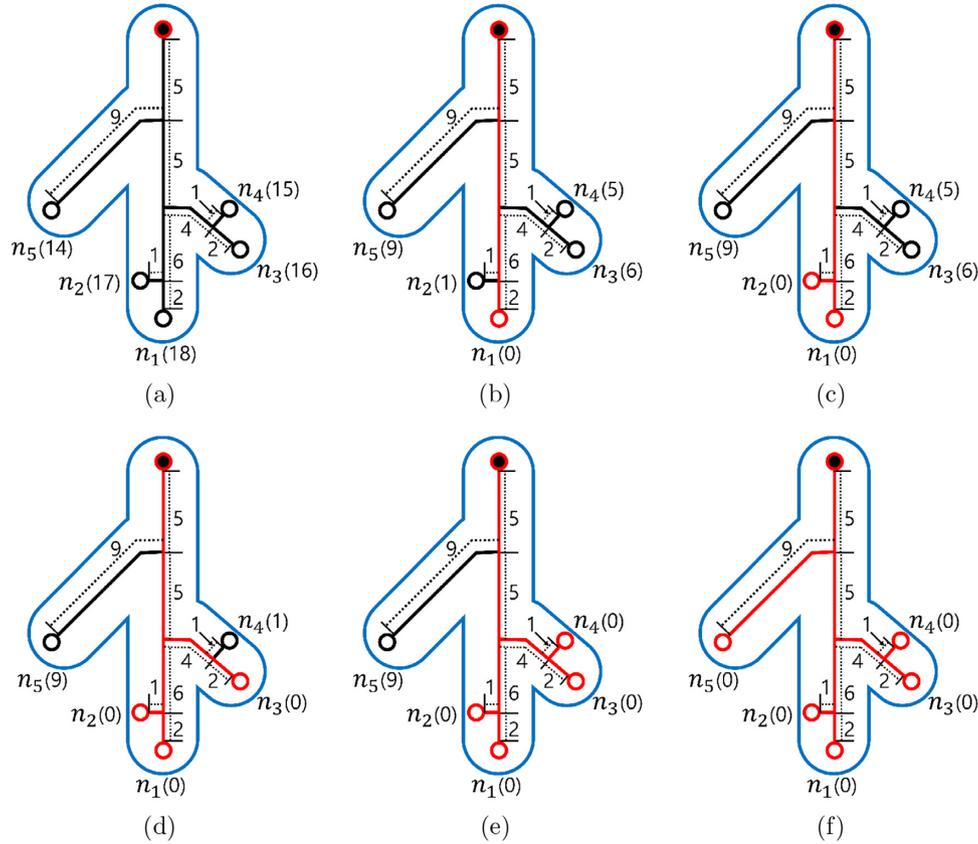


Figure A2: The computation of distance between trees for hairy node shaving. The number in a parenthesis denotes the path length from the leaf node to a subtree. (a) The initial tree T_0 only consists of a root node (red filled circle). Dist_0 is 80. (b) π_1 is concatenated (i.e. the red edge bounded by the red empty circle). Dist_1 is 21. (c) π_2 is concatenated. Dist_2 is 20. (d) π_3 is concatenated. Dist_3 is 10. (e) π_4 is concatenated. Dist_4 is 9. (f) π_5 is concatenated and no other paths remain. Dist_5 is 0.

Dist_i is a descriptor for the difference (or similarity) between two trees. If Dist_i decreases rapidly by concatenating π_i to T_{i-1} , we consider that π_i significantly contributes to T^{BWD} .

Definition 9. (Difference between distances) Let $\delta_i = \text{Dist}_{i-1} - \text{Dist}_i$ be the contribution of π_i to T^{BWD} .

Dist_i can be computed as follows. We create an ordered queue Q^{Shave} that stores all paths π_i in the non-increasing order of path length. Given T_{i-1} , we remove the first element π_i of Q^{Shave} and find its maximal subset that has null intersection with T_{i-1} . Each element removed from Q^{Shave} takes $O(k)$ where k represents the number of links in the path that do not belong to T_{i-1} . Hence, all elements in Q^{Shave} take $O(|N|)$ for $|N|$ nodes in T^{BWD} . We mark the nodes and links of T^{BWD} corresponding to π_i when it is concatenated.

Lemma 13. $\text{Dist}_{i-1} > \text{Dist}_i$.

Proof: Suppose T^{BWD} has m leaf nodes. $\text{Dist}_{i-1} > \text{Dist}_i$ because $\text{Dist}_{i-1} = \sum_{j=i}^m \lambda_{i-1,j} > \sum_{j=i+1}^m \lambda_{i-1,j} > \sum_{j=i+1}^m \lambda_{ij} = \text{Dist}_i$. \square

The strong monotonicity stated by Lemma 13 implies that, as more paths are concatenated, the being-concatenated tree is closer to the entire T^{BWD} . Hence, δ_i is always positive. The properties of Dist_i and δ_i are of interest. Figure A1a shows the profile of Dist_i for the paths in T^{BWD} (thus the entries in Q^{Shave}). The horizontal axis denotes the sequence of each path removed from

Q^{Shave} and the vertical axis denotes the distance of the intermediate tree. Recall that this distance is a measure of the difference between the two trees. The sudden decrease at a certain horizontal location corresponds to a significant branch in CA and is clearly detected by the sharp peaks in Fig. A1b. Figure A1c shows the close-up of Fig. A1b with a cutoff threshold value δ_{\min} indicated. The peaks higher than δ_{\min} produce recognized branches: Those under δ_{\min} are ignored.

Figure A1 is useful in that it can be used to choose the threshold δ_{\min} that determines the structure of medial axis \mathcal{MAT} by the decision whether a path π_i is concatenated or not. The choice of δ_{\min} is critical for the success of the algorithm. In Model-13, we used $\delta_{\min} = \sum_{i=1}^n \frac{\delta_i}{n}$ (the green horizontal line in Fig. A1c).

Figure A2 shows an example of the hairy node shaving process. Figure A2a shows T^{BWD} that was produced by the backward Dijkstra pass and T_0 , the root node of T^{BWD} (shown as filled red circle). There are five leaf nodes (shown as unfilled white circles). Note that $\lambda_{0,j}$ is given in the parenthesis. For example, $\lambda_{0,1}$ is given as 18 (= 5+5+6+2) and $\lambda_{0,2}$ is 17 (= 5+5+6+1).

T_0 consists of the root node of T^{BWD} . Hence, $\text{Dist}_0 = 80$ (= 18+17+16+15+14). Then, π_1 is removed from Q^{Shave} and concatenated to T_0 to make T_1 , i.e. $T_1 = \pi_1 \oplus T_0$ as shown in the red subtree in Fig. A2b. Note the updated statistics, i.e. $\lambda_{1,2} = 1$, $\lambda_{1,3} = 6$ (= 4+2), etc. and Dist_1 is 21 (= 1+6+5+9).

In the next step, π_2 is removed to be concatenated to T_1 to make T_2 , i.e. $T_2 = \pi_2 \oplus T_1$ as shown in the red subtree in Fig. A2c. Note that all statistics remain identical except $\lambda_{2,2}$

that is changed to zero from one. $\text{Dist}_2 = 20 (= 6+5+9)$. In the next step, π_3 is removed to be concatenated to T_2 to make T_3 , i.e. $T_3 = \pi_3 \oplus T_2$ as shown in the red subtree in Fig. A2.d. Note that $\lambda_{3,3}$ and $\lambda_{3,4}$ are changed to zero and one, respectively, while $\lambda_{3,5}$ remains unchanged. $\text{Dist}_3 = 10 (= 1+9)$. Figure A2.e and f shows the following steps similarly constructing T_4 and T_5 , respectively. Observe that $\text{Dist}_4 = 9$ but $\text{Dist}_5 = 0$ because $T_5 \equiv T^{\text{BWD}}$.

We observe δ_1 through δ_5 (see Definition 9): $\delta_1 = 59$, $\delta_2 = 1$, $\delta_3 = 10$, $\delta_4 = 1$, and $\delta_5 = 9$. Given these values, if we decide to remove the subtrees with contribution less than 5 (i.e. $\delta < 5$), we get the shaved tree T^{Shaved} that consists of three paths π_1 , π_3 , and π_5 because $\delta_1 = 59$, $\delta_3 = 10$, and $\delta_5 = 9$, respectively.

Algorithm 3 summarizes the algorithm to shave hairy nodes. Lines 1 through 5 compute the path from the root node to each leaf node and store each path with its corresponding leaf node in the ordered queue Q^{Shaved} . After the backward Dijkstra pass for the outrageous node removal, the paths of the remaining nodes (i.e. hairy nodes) are modified with its length changed. Hence, Steps 2 and 3 are necessary: Step 2 is done by simply following the traversal from a leaf node to the root node by following the parent of each node. Lines 6 through 9 compute Dist_i , together with δ_i , of T_i which is constructed by concatenating each path. Lines 11 through 15 construct the shaved adjacency tree. As stated earlier, Algorithm 3 takes $O(|N|)$ time in the worst case where $|N|$ is the number of nodes in T^{BWD} .

Algorithm 3: Shaving Hairy Nodes

input : Unshaved adjacency tree T^{BWD}

output: Shaved adjacency tree T^{Shaved}

```

1 for each leaf node  $n_i$  in  $T^{\text{BWD}}$  do
2   Get a path  $\pi_i$  from the root node to  $n_i$ ;
3   Calculate the path length for  $\pi_i$ ;
4   Push  $\pi_i$  into the ordered queue  $Q^{\text{Shave}}$  using a path
   length as a key;
5 end
6 while  $Q^{\text{Shave}}$  is not empty do
7   Remove the first element  $\pi_i$  from  $Q^{\text{Shaved}}$ ;
8    $T_i = T_{i-1} \oplus \pi_i$ ;
9   Compute  $\text{Dist}_i$  and  $\delta_i$ ;
10 end
11 Initialize  $T^{\text{Shaved}} \equiv \emptyset$ . for each path  $\pi_i$  do
12   if  $\delta_i \geq \delta_{\min}$  then
13     Concatenate  $\pi_i$  to  $T^{\text{Shaved}}$ ;
14   end
15 end

```

Steps 2 and 3 can be performed by simply following the path in the tree. Remember this is a tree. So, simply traversal to the parent from each node will do the job.

Appendix B: Determination of θ_{\max}

In Fig. B1, the red bars denote the frequency distribution of nodal angles in LCA. We observe that the distribution is right-skewed in the non-negative space and found that the log-normal distribution, shown as the black curve, fits it fairly well. We estimated its two parameters $\hat{\mu}$ and $\hat{\sigma}$ as 2.86 and 0.78 through the maximum likelihood estimation, respectively. Note that the blue bars correspond to the nodal angles of RCA.

We tested three values for θ_{\max} , 99, 95, and 90th percentiles from the log-normal distribution, which correspond to 107.5, 47.5, and 63.1 degrees of nodal angles, respectively. Note that if we choose ξ -th percentile, $(100-\xi)\%$ of the statistics over ξ is removed. Figure B2 shows a straightening example of RCA. Figure B2a and b shows T^{Shaved} before and after the straightening operation with $\theta_{\max} = 90\%$, respectively. We observe major changes in several locations. For example, see the black box. Figure B2c shows the close-up of the black box in Fig. B2a. Compare it with Fig. B2f that shows the straightened result with $\theta_{\max} = 90\%$: The seemingly self-intersection structure disappears after the straightening operation. In addition, we can easily notice several small improvements. Figure B2d and e shows the straightening results for 99th and 95th percentile, respectively. See the improvements with different thresholds. We use the straightened T^{Shaved} as \mathcal{MAT} . In the case of RCA, the 2,252 nodes in the T^{Shaved} before straightening decreased to 2,241, 2,158, and 2,066 nodes for the 99, 95, and 90th percentile thresholds, respectively.

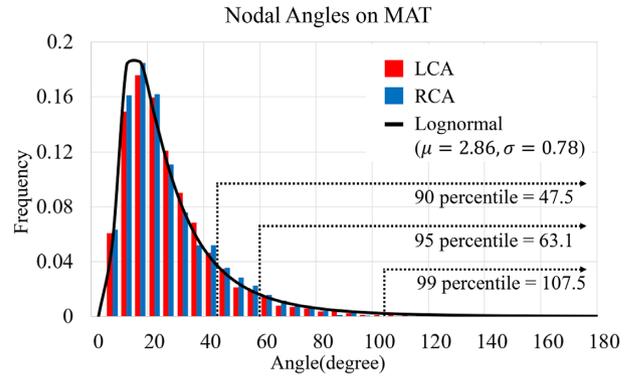


Figure B1: The frequency distribution of nodal angles in T^{Shaved} . The red bars denote the frequency distribution of LCA. The blue bars correspond to the nodal angles of RCA. The black curve shows the estimated log-normal distribution.

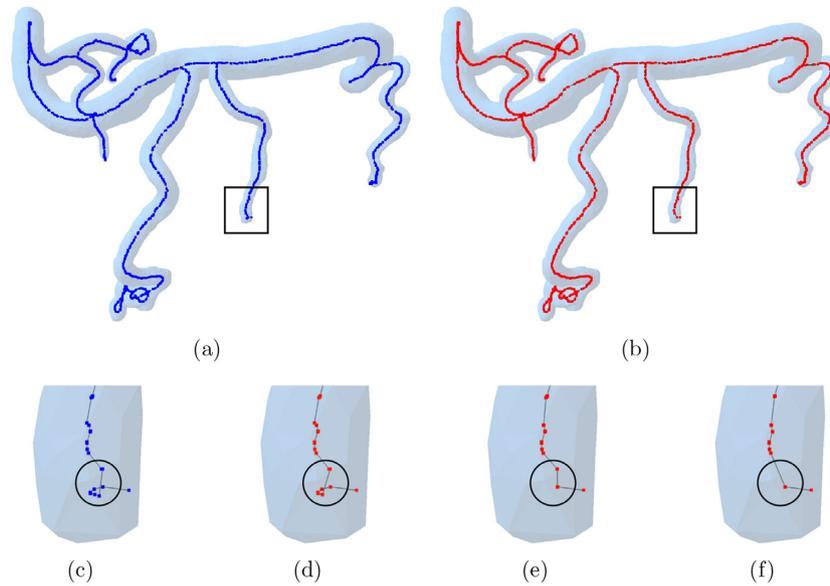


Figure B2: The straightening result for different threshold values. (a) T_{shaved} . (b) MAT with 90th percentile (47.5 degree). (c) A close-up of (a) in the black box. (d) Straightening result with 99th percentile (107.5 degree). (e) Straightening result with 95th percentile (63.1 degree). (f) Straightening result with 90th percentile (47.5 degree).

Appendix C: Topological Handles in Test Cases

Figure C1a–c shows the topological handles in the CA. We believe that these artifacts were created by the measurement error during the CT scanning in conjunction with an inappropriate decision-making in the mesh extraction algorithm of the program used. Most left ventricles have nonzero gs . See Fig. C1d

and e for tiny artifacts that might be caused by measurement error. Figure C1e is Model-13, which is used throughout this paper: See the tiny holes in the red circle in Fig. 2e. It is interesting that LV of Model-15 has $g = 333$ (see Fig. C1f). The low resolution of the CT-scanning techniques and possible bugs in mesh generation program might be responsible for such errors.

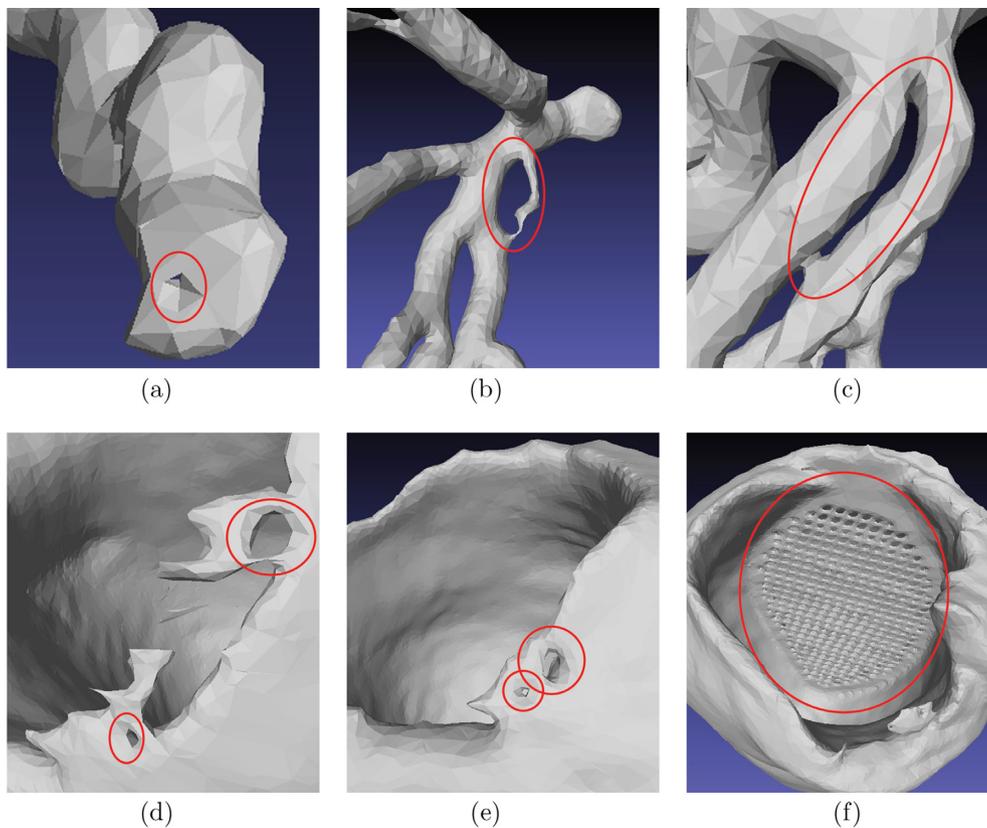


Figure C1: The example of holes among the models. Visualization is done via MeshLab (Cignoni et al. 2008). (a) CA in Model-1. (b) CA in Model-2. (c) CA in Model-9. (d) LV in Model-12. (e) LV in Model-13. (f) LV in Model-15.

Appendix D: Summary of 20 Test Cases

Table D1: Statistics for the 20 Samsung models. Each model consists of three components: LV, CA1, and CA2. The row EC shows the Euler characteristic $|V| - |E| + |F| - 2$. g is the genus.

Model ID		1	2	3	4	5	6	7	8	9	10
LV	V	17,929	19,215	23,278	27,568	29,849	30,754	30,970	31,509	31,749	33,032
	E	53,781	57,639	69,834	82,698	89,541	92,268	92,910	94,533	95,241	99,102
	F	35,854	38,426	46,556	55,132	59,694	61,512	61,940	63,022	63,494	66,068
	EC	0	0	-2	0	0	-4	-2	-4	0	-4
	g	0	0	1	0	0	2	1	2	0	2
CA1	V	4,633	4,604	3,646	2,938	6,760	4,027	3,618	5,288	6,071	6,231
	E	13,899	13,806	10,932	8,808	20,274	12,075	10,848	15,858	18,213	18,687
	F	9,266	9,204	7,288	5,872	13,516	8,050	7,232	10,572	12,142	12,458
	EC	-2	0	0	0	0	0	0	0	0	-2
	g	1	0	0	0	0	0	0	0	1	0
CA2	V	6,863	4,089	4,038	4,593	7,181	5,875	5,107	4,933	5,925	9,358
	E	20,583	12,273	12,108	13,773	21,537	17,619	15,315	14,793	17,769	28,068
	F	13,722	8,182	8,072	9,182	14,358	11,746	10,210	9,862	11,846	18,712
	EC	0	-4	0	0	0	0	0	0	0	0
	g	0	2	0	0	0	0	0	0	0	0
Model ID		11	12	13	14	15	16	17	18	19	20
LV	V	33,745	33,943	34,642	36,407	39,773	41,694	41,783	41,907	46,090	48,720
	E	101,235	101,841	103,950	109,215	121,311	125,082	125,349	125,727	138,318	146,154
	F	67,490	67,894	69,300	72,810	80,874	83,388	83,566	83,818	92,212	97,436
	EC	-2	-6	-10	0	-666	-2	-2	-4	-18	0
	g	1	3	5	0	333	1	1	2	9	0
CA1	V	5,280	5,063	5,928	4,891	4,393	3,665	4,141	4,410	4,653	5,932
	E	15,834	15,183	17,778	14,667	13,173	10,989	12,417	13,224	13,953	17,790
	F	10,556	10,122	11,852	9,778	8,782	7,326	8,278	8,816	9,302	11,860
	EC	0	0	0	0	0	0	0	0	0	0
	g	0	0	0	0	0	0	0	0	0	0
CA2	V	4,436	6,975	8,964	6,461	8,300	6,805	7,309	7,728	7,570	8,099
	E	13,302	20,925	26,886	19,377	24,894	20,409	21,921	23,178	22,704	24,291
	F	8,868	13,950	17,924	12,918	16,596	13,606	14,614	15,452	15,136	16,194
	EC	0	-2	0	0	0	0	0	0	0	0
	g	0	1	0	0	0	0	0	0	0	0

Appendix E: Summary of Acronyms

Table E1: Summary of the acronyms in the manuscript.

Acronym	Meaning
LV	Left ventricle
CA	Coronary artery
LCA	Left CA
RCA	Right CA
CT	Computed tomography
MAT	Medial axis transformation
CDT	Constrained Delaunay triangulation
VD	Voronoi diagram