Eurographics Workshop on Visual Computing for Biology and Medicine (2021)
S. Oeltze-Jafra, N. N. Smit, and B. Sommer (Editors)

Automatic Cutting and Flattening of Carotid Artery Geometries

P. Eulzer[1], K. Richter[2], M. Meuschke[1,3], A. Hundertmark[2] and K. Lawonn[1]

[1]University of Jena, Faculty of Mathematics and Computer Science, Germany
[2] University of Koblenz-Landau, Institute for Mathematics, Germany
[3] University of Magdeburg, Department of Simulation and Graphics, Germany

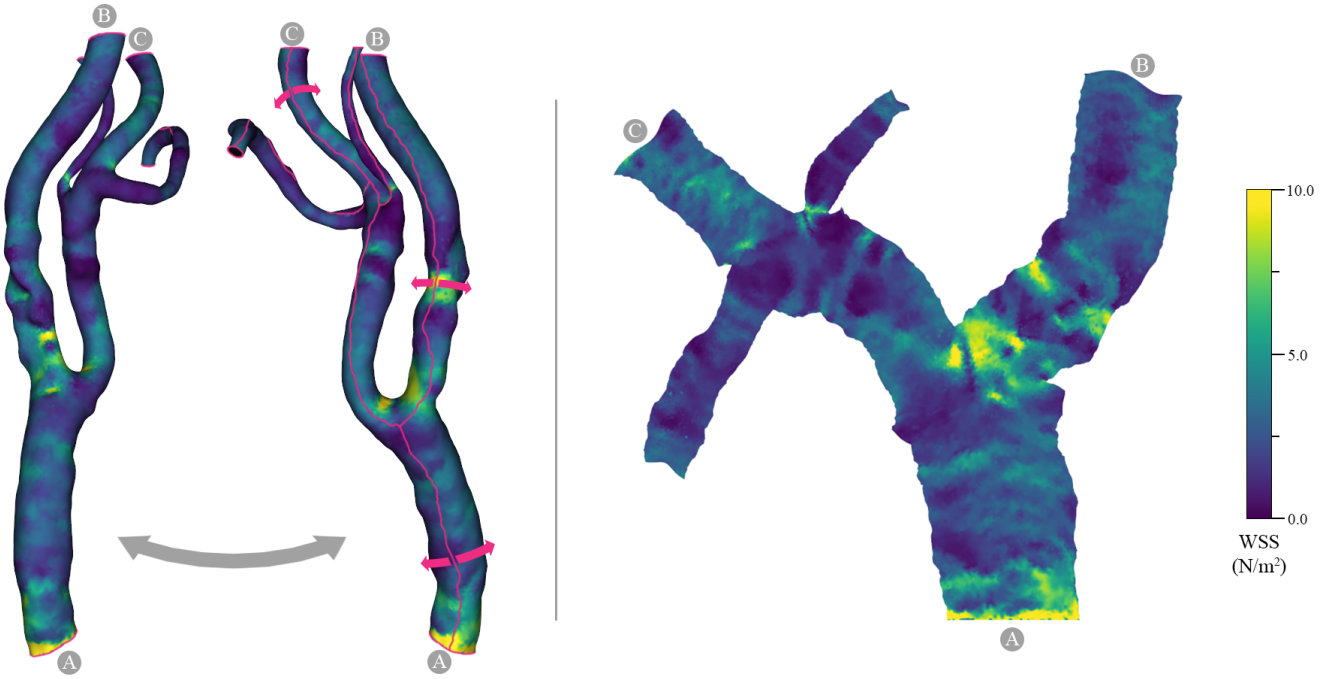


Figure 1: *A surface parameter (wall shear stress) is colormapped to a carotid artery. In a 3D projection (left) interaction is necessary to examine the whole surface. We propose an automatic method to cut and flatten vessel trees, which shows the full surface while retaining its proportions and layout (right). A: common carotid. B: internal carotid. C: external carotid.*

Abstract

We propose a novel method to cut and flatten vascular geometry that results in an intuitive mapping between the 3D and 2D domains. Our approach is fully automatic, and the sole input is the vessel geometry. We aim to simplify parameter analysis on vessel walls for research on vascular disease and computational hemodynamics. We present a use case for the flattening to aid efforts in investigating the pathophysiology of carotid stenoses (vessel constrictions that are a root cause of stroke). To achieve an intuitive mapping, we introduce the notion of natural vessel cuts. They remain on one side of vessel branches, meaning they adhere to the longitudinal direction and thus result in a comprehensible unfolding of the geometry. Vessel branches and endpoints are automatically detected, and a 2D layout configuration is found that retains the original branch layout. We employ a quasi-isometric surface parameterization to map the geometry to the 2D domain as a single patch. The flattened depiction resolves the need for tedious 3D interaction as the whole surface is visible at once. We found this overview particularly beneficial for exploring temporally resolved parameters.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization;*** • ***Applied computing*** → *Life and medical sciences;* • ***Computing methodologies*** → *Mesh geometry models;*

DOI: 10.2312/vcbm.20211347

1. Introduction

Cardiovascular disease is the leading cause of mortality and morbidity worldwide [GBD16]. The majority of cases are attributed to ischemic heart disease and stroke [JLM*17]. These are typically caused by *stenosis*, a localized constriction of the arteries supplying the heart or brain. A common issue is stenosis in the carotid arteries due to atherosclerotic plaque buildup. The two internal carotids provide the majority of blood supply to the cranium. They split off the common carotid artery at the *carotid bifurcation*. Plaque can often be found at or after the bifurcation.

Recent advances in simulated hemodynamics allow for modeling of blood flow properties based on computational fluid dynamics (CFD) [CL13; SH18]. These simulations are performed on patient-specific arterial wall geometry. Typically, one flow inlet is defined, for instance, at a cross-section of the common carotid artery. Here, inflow is simulated following the pattern of a standardized human heart cycle. The flow exits the domain at one or more predefined outlets, which lie downstream from the inlet. Together with the wall, the inlet and outlets constitute boundary conditions that allow solving the internal flow field. This yields various parameters at arbitrarily high resolutions, some of which cannot be measured in-vivo. These simulation results can help understand the progression of cardiovascular disease and may even provide additional markers for medical treatment decisions in the future. A principal interest is the analysis of wall-related parameters, such as normal wall stress or wall shear stress (WSS), as they show how the vessel wall is affected by the flow field.

The analysis of CFD data is a cumbersome process. Often, a large spatiotemporal data space needs to be examined, where comparisons need to be made regarding values over time and different parameterizations of the simulation. To provide better insights into this complex data, a variety of visualization techniques and frameworks have been proposed for medical flow visualization [OMN*19]. Following this line of research, we generate overview depictions for at-a-glance assessments of wall-related CFD results. Existing model-based techniques for the flattening of vessel surfaces either create patches and do not show a continuous domain [AS04; BGP*11; GSK*12], or do not preserve the surface area [MK15]. Our flattening approach is a global surface parameterization that creates a single patch. The parameterization preserves the vertex and connectivity information (except for the cut), making it simple to map any data fields defined on vertices or faces of the mesh. For a realistic assessment of parameter distributions, we minimize area distortion while keeping the layout close to the original structure. Our approach is also fully automatic and requires no manual cuts or predefined landmarks. In contrast to other methods, the single input for our techniques is the 3D mesh of the vessel tree, we do not require a centerline. Further, our flattening results in intuitive 2D views of the unrolled inner vessel wall. We achieve this by introducing natural vessel cuts, which remain on one side of the cylindrical vessel shape. Our method supports vessel trees of various sizes with an arbitrary number of outlets, contrary to existing approaches to map the walls of the human carotid that only consider a limited region of the vessel with a specific geometry [CCR20; CES*08].

Furthermore, we investigate a use case for the flattening. We integrated the method into a CFD pipeline that simulates blood flow in the carotids of real patients. We provide a GUI to generate flattened depictions of the extracted wall geometries, on which scalar fields can be mapped. The tool supports navigation of the 3D and 2D spaces as well as the temporal dimension through interpolated animation of timesteps. The demonstration focuses on the carotid as an example of a highly relevant application area. However, we also show how our method can be generalized to any vessel tree.

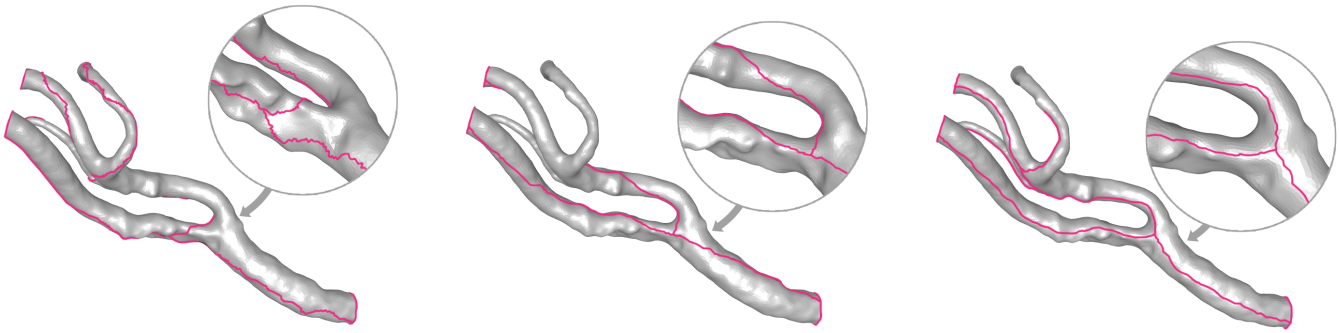
2. Related Work

Flattening spatial data is a common technique in medical visualization to reduce the perceived complexity of structures, as is reviewed in the survey by Kreiser et al. [KMM*18]. Methods have been proposed in a variety of scenarios where anatomical structures need to be visually analyzed, for example, the colon [MZGK11], bones [MPG*17], or heart valves [EEL*19; LER*20]. Usually, the objective is to create a 2D domain, which provides an overview that requires less interaction than a direct rendering of its 3D counterpart. These 2D visualizations often serve as a map to find points of interest and navigate the spatial and sometimes temporal data space. Flattened illustrations of arterial vessel trees were found superior to 3D surface renderings w.r.t. accuracy and efficiency when assessing a scalar field such as shear stress on the vessel wall [BGP*11].

A large group of vessel flattening approaches are point-based *reformation* techniques. Kanitsar et al. [KFW*02] introduced curved planar reformation, which operates directly on medical volume images and provides continuous views of vessels after the user defined a centerline. The technique was adapted and modified multiple times, resulting, for instance, in the centerline reformation method [MVB*12]. Mistelbauer et al. [MMV*13] also demonstrate how viewing directions can be aggregated into a single depiction. Diepenbrock et al. [DHS*13] use reformation to display PET activity in the carotids of mice.

If vessel walls were segmented, for example, to perform a CFD simulation of blood flow, the geometric model can be flattened directly. This flattened depiction can be used to visualize parameter distributions that occur on the surface, either measured or resulting from simulation. Neugebauer et al. [NGB*09] propose view-aligned 2D projections of scalar surface fields for aneurysm analysis. Their method, however, can only show small subsections of a vessel and is based on projection. It cannot flatten the whole vessel and does not preserve geometric properties like area or shape. Tao et al. [THQ*16] construct a full vessel map that provides a 2D overview of the internal flow field in a vessel. They use a voxelization of the 3D domain and transfer the voxels into a 2D graph. Since the voxelization does not provide a direct mapping of the wall it is not suited for analysis processes in medical research or clinical decision making where wall-related parameters are the main interest. Similarly, Pandey et al. [PSY*19] create a network visualization of cerebral artery trees and Eulzer et al. [EMKL21] proposed a map of the carotids with a standardized layout for the main and side branches. Both methods flatten the centerline, not the vessel wall surface.

For overviews of the mesh surface, model-based techniques are



(a) *Generic disk-cut.* **(b)** *Minimizing geodesic distance.* **(c)** *Natural vessel cuts (ours).*

Figure 2: *Different cutting approaches that all result in a mesh homeomorphic to a disk.*

best suited, which employ a surface parameterization that maps every 3D vertex to a 2D coordinate in the parameter domain. Ideally, such a mapping is bijective, i.e., every point on the 3D surface can be mapped to exactly one point in the parameter domain and vice versa. Surface parameterizations have been proposed for vessel trees or tubular surfaces, as the resulting 2D coordinates are useful for illustrative techniques like hatching, contouring, or integrating depth cues. Kälberer et al. [KNP11] proposed a method to generate 2D coordinates for stripe patterns on tree-like structures. The technique is, however, not suited for flattening as these mappings do not aim for bijectivity. Hence, a 2D rendering would suffer from extensive overlaps of branches. Lichtenberg and Lawonn [LL20] presented a similar approach. They compute the 2D coordinate of the first dimension on the 3D geometry and the second coordinate in screenspace. The resulting 2D coordinates suffice for illustrative purposes, but the technique cannot create a flattened depiction of the object. Surface parameterizations for flattening have been proposed for aneurysms explicitly [GSK*12; MVB*16], which requires a cut along the aneurysm base (ostium). These methods are not global w.r.t. the vessel, as they only map selected patches.

To flatten arbitrary vessel trees Borkin et al. [BGP*11] proposed 2D projections of cylindrical vessel segments. They map coronary artery trees to a diagram representation for overview assessment of simulated WSS. The geometry is not preserved as they use a projection and vessel bifurcations cannot be shown. Marino et al. [MK15] introduced a mapping for arbitrary branch configurations, which shows the whole vessel and achieves bijectivity. Based on the centerline, a planar configuration is found that mimics the 3D layout of the current view direction. The boundary is fixed around this configuration showing the vessel diameter. The mesh is cut from root to leaves over branch nodes. Then, the surface is fitted into the fixed boundary shape with an adapted harmonic mapping. For our purposes, however, the method is not suited as the area is not preserved. The fixed boundary encodes the vessel diameter and is, therefore, necessarily smaller than a path that would follow the circumference. The layout is also strictly dependent on the viewing angle. Furthermore, the cuts optimize geodesic distance, which results in unintuitive cutting paths, as we show in Section 4.

Area-preserving maps of vessel trees have also been proposed. Zhu et al. [ZHT05] first showed a technique for geometric cuts and unrolling of Y-shaped tubular regions, including an optional step to minimize area distortion. Several of such regions can be stitched to represent a vessel tree. They use conformal maps that may result in unexpected cuts. The method was adapted by Chiu et al. [CES*08], who reconstruct the inner and outer walls of the carotids from ultrasound images. The 3D surfaces are subjected to an area-preserving flattening that is used to visualize vessel thickness, which is an indicator for atherosclerosis. Antiga et al. [AS04] created a standardized 2D map of the carotid bifurcation, where three topological cylinders are found, cut, and unrolled in three individual planar maps. As these maps suffer from discontinuities where the tubular segments would join, an L-shaped map of the carotid bifurcation was proposed, which then helped to develop biomarkers for plaque progression/regression in clinical trials [CLC13; CCC16; CUS*17]. Recently, Choi et al. [CCR20] minimized the area distortion within the L-shape. The fundamental problem of these approaches is that they are specific to a limited segment of the carotid, whose geometry is well-defined by three connected cylinders: the common, internal, and external carotid. The section is about 20-25 mm in length, which allows to assume the branches as straight and cut them with predefined planes along known landmarks. On average, the carotid covers more than 200 mm in adults [CGRH16]. In the L-shaped map stenoses that occur anywhere but directly at the carotid bifurcation will be hidden. Furthermore, for accurate CFD simulations, larger segments with potentially more than two outlets are of interest.

3. Objectives

Our goal is to create a flattening through surface parameterization. We assume the vessel tree is a triangulated surface with vertices $\mathcal{V}$ and faces $\mathcal{F}$, where $(i,j,k) \in \mathcal{F}$ with $i,j,k \in \mathcal{V}$ means that the vertices i,j,k form a triangle. The 3D position of a vertex i is denoted by $\mathbf{v}_i \in \mathbb{R}^3$. We want to find a configuration of (u,v) parameter coordinates where each vertex $\mathbf{v}_i = (x_i, y_i, z_i)$ corresponds with exactly one point (u_i, v_i). For an unambiguous representation of scalar field data on the flattened surface, we aim for a bijective mapping between the 3D and parameter space. No overlaps or triangle flips should occur.

We intend to facilitate use cases in CFD and medical research, where surface attributes on the vessel need to be analyzed. This

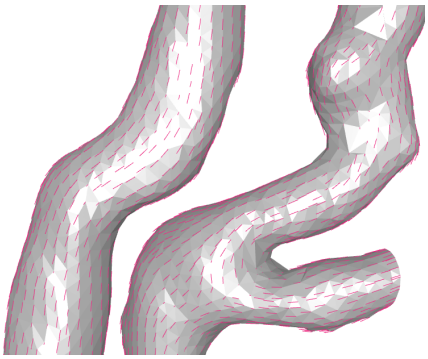


Figure 3: *The minimal principal curvature is aligned with the vessel direction, serving as the basis to orient the natural vessel cuts.*

analysis involves localization of certain attribute distributions to understand, e.g., where irregularities occur. Naturally, not only the location but also the size of these feature distributions needs to be accurately portrayed for visual analysis. Therefore, the parameterization needs to preserve surface area if possible. Simultaneously, the surface layout should be intuitive, i.e., it should be easy to recognize anatomical features and mentally project them into the 3D space. All parts of the input surface should be visible, even if in specific cases some side-branches might be less medically relevant. Mapping the total surface per default makes the technique more generally applicable. Also, for CFD research it is important to examine and compare the behavior of parameters on the whole surface that defines the simulation domain.

We expect the input mesh to be triangulated and assume open boundaries at inlets and outlets, which is the typical morphology of vessel walls in CFD blood flow simulations. The rest of the mesh should be watertight and not contain self-intersections. The vertex indices of each triangle must be ordered consistently with normals pointing outwards. As we focus on vasculature, we assume a tree-like surface structure where sub-branches on average get thinner as compared to their root branch. Result data can be given as scalar/vector fields discretized on $\mathcal{V}$ or $\mathcal{F}$. It should be kept in mind that vessels do not consist of perfect cylinders and small deformations are the norm. Also, stenoses are possible that can disturb the cylindrical shape.

4. Cutting

To find a valid surface parameterization that can be rendered in a flat layout, the input mesh must be cut such that it becomes homeomorphic to a disk. We want to find *natural* vessel cuts that are necessary for an intuitive flattened layout. We define natural as follows:

1. The first cut starts at the inlet boundary of the vessel tree.
2. At each branch of the vessel tree, the cut splits in two child cuts, one for each branch.
3. At each outlet boundary exactly one cut terminates.
4. Cuts may only intersect at the point where they connect.
5. Wherever possible, cuts travel in the longitudinal direction of cylindrical vessel segments.

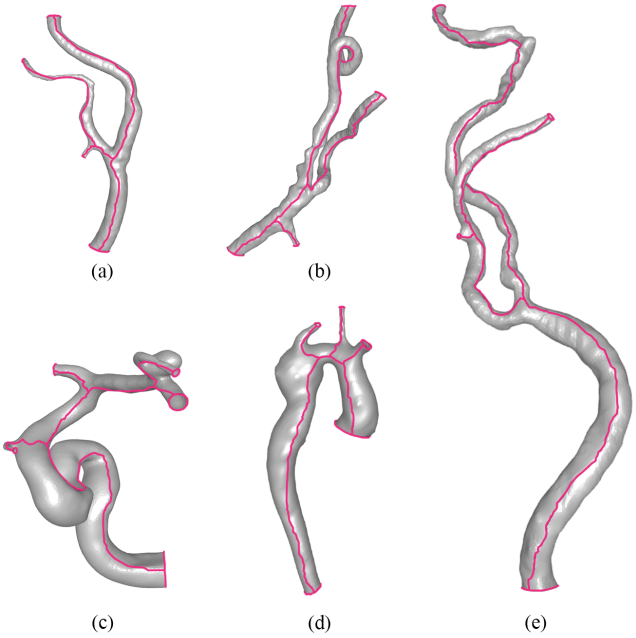


Figure 4: *Natural vessel cuts on different geometries. (a)-(b) Human carotid arteries. (c) Cerebral artery with aneurysm. (d) Aorta with aneurysm. (e) Large segment of a carotid.*

This results in a minimal number of connected cuts that slice open each vessel segment, creating a single surface patch. The definition is similar to the boundary creation as described by Marino et al. [MK15]. However, we additionally impose the last point as a crucial condition. Cuts should remain on one side of the cylinders and not wrap around them. By taking this requirement into account, we are able to generate cuts that enable intuitive mental mapping between 3D and 2D, as the surface will "unfold" in a consistent orientation. The advantage of these cuts becomes apparent when compared to the traditional approaches of either using a generic cutting algorithm [GGH02] or minimizing geodesic distance (see Figure 2). We create natural vessel cuts by following the minimal principal curvature, which, on a cylindrical object, defines a tangent field along its length.

4.1. Sorting boundaries

To create the cuts we first find the set of existing boundary vertices $\mathcal{V}_\mathcal{B} \subset \mathcal{V}$. We select an arbitrary vertex in $\mathcal{V}_\mathcal{B}$ and recursively search for one neighboring vertex that is also in $\mathcal{V}_\mathcal{B}$ until we found a loop. Then, the process is repeated with an unvisited boundary vertex until all boundary vertices are visited. This gives us one ordered boundary loop for the inlet and one for each outlet. To prepare for the flattening where these loops are effectively "unrolled", we check that the loop orientation is consistent with the winding order of the mesh faces. This means, a vertex candidate is only appended to a loop if it and the previous vertex would occur in the same order as they do in the face they share. They can only share one face as they are both on the boundary. This condition only needs to be tested for the second vertex of a new loop, as two neighbors to the

first vertex are possible candidates. Ultimately, we sort the list of loops by the arc length of the loops. We expect the root of the vessel tree to lie at the thickest branch. Hence, the longest boundary loop is considered to be the inlet (cf. Figure 1A), all others are considered to be outlets.

4.2. Aligning curvature

We compute the principal curvature directions at each vertex and extract the minimum principal curvature $\mathcal{C}_{\min} \subset \mathbb{R}^3$, which is a good geometric estimator of the longitudinal direction along a cylinder. The idea was also employed by Lichtenberg and Lawonn [LL20] to compute a continuously increasing scalar field across a vessel tree. We use the method by Panozzo et al. [PPR10] to compute the principal curvature cross-field. The principal curvature is analytically solved on a best-fit quadric over the neighborhood of each vertex. This method is fast and robust. It allows to define a scale for the quadric based on the average edge length of the mesh edges e. As in the original proposal, we set the scale to a default of $5e$. Then, the quadric fitting will take into account a large portion of neighboring vertices, which effectively smoothes noise and parts of the surface that deviate from the ideal cylinder. As such, the curvature field will be aligned with the overall shape of the vessel tree. It is robust against irregularities in the vessel structure, small aneurysms, and deformations from stenoses.

A problem when using principal curvature is that the sign of individual elements $\mathbf{c}_i \in \mathcal{C}_{\min}$ is ambiguous. At each vertex $\mathbf{c}_i$ or $-\mathbf{c}_i$ can be computed. We need all directions in $\mathcal{C}_{\min}$ to be aligned towards the root of the tree geometry. We resolve this ambiguity as proposed by Lichtenberg and Lawonn [LL20]. First, the geodesic distance $\mathcal{G} \subset \mathbb{R}$ is computed as a scalar field over $\mathcal{V}$, where each vertex is assigned the distance to the inlet boundary. We use the geodesics in heat method [CWW13] as it is fast and sufficient for the purpose. As $\mathcal{G}$ increases strictly monotonously over the vessel tree, we can use its gradient at vertex positions $\mathbf{g}_i \in \nabla\mathcal{G}$ to obtain the aligned curvature $\mathcal{Z}$:

$$\mathbf{z}_i = \begin{cases} \mathbf{c}_i, & \text{if } \langle \mathbf{c}_i, \mathbf{g}_i \rangle < 0 \\ -\mathbf{c}_i, & \text{otherwise.} \end{cases} \tag{1}$$

The resulting field of aligned minimal curvature is shown in Figure 3.

4.3. Finding optimal paths

The mesh is now sliced open with natural vessel cuts that when traced from outlets to the inlet minimize deviation from the directions given in $\mathcal{Z}$. We will use cuts that run along existing mesh edges, i.e., no new topology will be introduced in the process. This makes it simple to map any data fields between the 3D and 2D representations, as a complete set of (u,v) coordinates will be defined.

Picking any two vertices on the mesh, an optimal path can be found between them by interpreting the mesh as a directed graph. In this graph, a unique node exists for each element in $\mathcal{V}$ and each mesh edge is represented by two directed graph edges, one in each direction. By associating each directed edge with a travel cost, we can find the path between two selected nodes that minimizes the total cost. For simplicity, we use Dijkstra's algorithm [Dij59] to find the path. Due to its logarithmic complexity it also scales with large meshes. The cost function can be used to control the path. From a mesh vertex $\mathbf{v}_i$ edges exist to any direct neighbor $\mathbf{v}_j$. When setting the cost to the Euclidean distance from $\mathbf{v}_i$ to $\mathbf{v}_j$, the path with minimal geodesic length is found:

$$E_g(\mathbf{v}_i, \mathbf{v}_j) = ||\mathbf{v}_j - \mathbf{v}_i||. \tag{2}$$

In this case, the graph could be undirected. To minimize the deviation of the path from the aligned curvature $\mathbf{z}_i$ at $\mathbf{v}_i$, however, we need directed edges with costs:

$$E_c(\mathbf{v}_i, \mathbf{v}_j) = 1 - \left\langle \frac{\mathbf{v}_j - \mathbf{v}_i}{||\mathbf{v}_j - \mathbf{v}_i||}, \frac{\mathbf{z}_i}{||\mathbf{z}_i||} \right\rangle \tag{3}$$

Due to the normalization the edge length is ignored. If edge lengths deviate strongly from their mean, it is possible to factor in higher costs for longer edges by interpolating the geodesic and curvature cost:

$$E_{cg}(\mathbf{v}_i, \mathbf{v}_j) = a \cdot E_c(\mathbf{v}_i, \mathbf{v}_j) + (1-a) \cdot E_g(\mathbf{v}_i, \mathbf{v}_j) \tag{4}$$

This is usually not a problem for meshes used in CFD, which typically have a semiregular triangulation. Therefore, we found the most intuitive cuts ensue from using E_c alone (equivalent to $a = 1$) but reserve the option to use E_{cg} if required.

After creating the graph representation with edge costs, we compute one cut for each outlet. We resolve the ambiguity of where to split the path at branch locations by creating the cuts in reverse, i.e., we start one cut at each outlet. The first cutting path runs from the largest outlet to the inlet. This initial cut follows the vessel branches with the largest diameter and, therefore, the highest flow volume. On this principal branch the cut will lie as straight on the vessel geometry as the triangulation allows (cf. Figure 4). Every vertex that lies on a cut is appended to a set $\mathcal{V}_C \subset \mathcal{V}$. For the following cutting paths we iterate the remaining outlet boundaries from largest to smallest. In every iteration we find the vertex in $\mathcal{V}_C$ that is closest to the starting outlet. Choosing this vertex as the target ensures that we always hit an existing cut in vicinity of a branch location and that cuts do not cross.

We provide two options to decide which point on an inlet/outlet boundary will be chosen to start or end the cut. One, the points on a boundary loop can be connected to a common imaginary node with zero edge costs. The imaginary node is not present on the mesh and only used in the graph search. It has only outgoing edges at an outlet and only incoming edges at the inlet. When computing a cut, the imaginary nodes ensure that the most cost-efficient path is found. This requires no further input and is fully automatic. If, however, a cut should face the observer (which makes it easier to mentally reconstruct the unfolding process), a single 3D point can be given as a parameter, for example, the current viewpoint or a fixed location in front of the vessel. Then, we choose the point with minimal Euclidean distance to this location from each boundary loop. Figure 4 shows the robustness of this approach with different vessel geometries. Finally, the cut is applied to the mesh by duplicating the vertices in $\mathcal{V}_C$ and solving the topology in $\mathcal{F}$ accordingly. If data fields are present on the affected vertices they are copied as well.

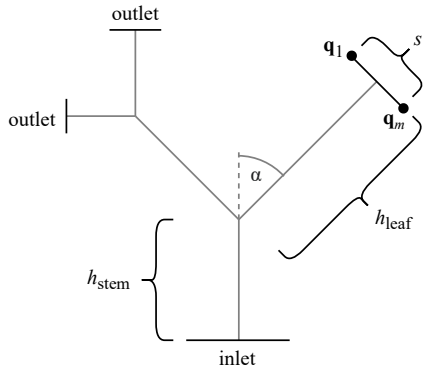


Figure 5: *Possible layout initialization that fixes the (u,v) coordinates of inlet and outlet boundaries to resemble a tree. The angle is set to $\alpha = 45°$ but can be arbitrarily chosen. All other values are read from the cuts.*

5. Flattening

The flattening should balance area preservation and recognizability of the anatomy. We use a boundary-free approach, which benefits both requirements. However, we fix the inlet boundary loop to an iso-u line centered on $(u,v) = (0,0)$, which serves as a reference in parameter space above which the remaining coordinates can develop. Fixing the inlet this way increases the comparability of different flattenings. We use the as-rigid-as-possible (ARAP) method for mesh deformation introduced by Sorkine and Alexa [SA07], which was extended for single-patch surface parameterization by Liu et al. [LZX*08]. The ARAP method attempts to retain distances in triangles, preserving their shape up to a rigid transformation, which yields quasi-isometric results when applied to surface parameterization. A trade-off is achieved between area preservation and conformality. The drawback of this method is that it requires an initialization of (u,v) coordinates in parameter space. Typically, this initialization can be done by mapping the surface to a convex shape, such as a square or circle, for which fast and robust methods exist, e.g., harmonic parameterization. We found, however, that the tree-like structure of the mesh is not suited for this approach and extensive overlaps occur within the parameter domain. A better way is to provide a flat initialization that already resembles the expected target layout and then use ARAP deformation to restore the proportions as observed on the 3D surface.

Different ideas have been proposed for the layout of vessel maps, depending on the use case. Recently, Eulzer et al. [EMKL21] showed a map of carotid centerlines where the most important segments, the common and internal carotid, are straightened to a line. Side branches are laid out at a fixed angle. We build on this idea, which simplifies the sometimes convoluted 3D vessel shape. We fix the inlet and outlets to predetermined (u,v) coordinates and move them to their desired locations, as shown in Figure 5. Each inlet/outlet loop contains an ordered set of boundary vertices $\mathbf{p}_1,...,\mathbf{p}_n \in \mathbb{R}^3$. We re-order the loops such that they start at the cut and the last vertex $\mathbf{p}_n$ is the copy created of $\mathbf{p}_1$ during the cutting. The loops are unrolled as straight lines, where each boundary point $\mathbf{p}_i \in \mathbb{R}^3$ is associated with a point in parameter space $\mathbf{q}_i \in \mathbb{R}^2$. Given the arc length of one such loop:

$$s = \sum_{i=2}^{n} ||\mathbf{p}_i - \mathbf{p}_{i-1}||, \tag{5}$$

we create an intermediate initialization of the mapping that preserves the arc length:

$$\tilde{\mathbf{q}}_i = \begin{cases} \begin{pmatrix} -0.5s \\ h_{\text{leaf}} \end{pmatrix} & \text{if } i = 1 \\ \tilde{\mathbf{q}}_{i-1} + \begin{pmatrix} ||\mathbf{p}_i - \mathbf{p}_{i-1}|| \\ 0 \end{pmatrix} & \text{otherwise.} \end{cases} \tag{6}$$

The resulting line is centered around the v-axis in parameter space. The height h_{leaf} approximates the length of the vessel segment that is capped by the boundary loop (cf. Figure 5). The intermediate mapping is then recursively displaced by the transforms $\Phi_1,...,\Phi_d$, where d is the branch depth of an outlet:

$$\mathbf{q}_i = \Phi_1 \circ ... \circ \Phi_d(\tilde{\mathbf{q}}_i), \qquad \Phi(\tilde{\mathbf{q}}_i) = \mathbf{R}\tilde{\mathbf{q}}_i + \begin{pmatrix} 0 \\ h_{\text{stem}} \end{pmatrix}. \tag{7}$$

This is effectively a classical transformation hierarchy, where the displacement of a segment is passed on to all child segments. The 2×2 matrix $\mathbf{R}$ describes a rotation around an angle α. The angle can be constant or vary depending on the geometry or desired layout. We automatically determine the sign of α by testing if the branch lies to the left or right of the stem it is attached to. A good approximation for the heights h_{stem} and h_{leaf} can be read from the scalar field $\mathcal{G}$ by subtracting the value at the start from the value at the end of the corresponding vessel cut. Note that slight variations of the heights do not affect the result, as the (u,v) coordinates will be displaced in the ARAP stage anyway. The final parameter coordinates of this step only serve as an initialization.

After the boundary loops are fixed to the (u,v) plane, the remaining surface coordinates need to be mapped to the parameter domain. It is not important which algorithm is used for this purpose, as long as it creates a bijective mapping that is adequate for use with the ARAP method. As originally proposed [LZX*08], we use least squares conformal maps (LSCM) [LPRM02] for this step. LSCM requires at least two fixed (u,v) coordinates but does not need a fixed boundary. We solve the LSCM with fixed inlet/outlets, which results in a flattening that already approximates the desired layout (see Figure 6). Finally, the (u,v) coordinates are optimized with the ARAP method, which displaces the 2D mesh such that it retains the proportions of its 3D counterpart wherever possible (a quasi-isometric parameterization). In this last step, only the inlet boundary remains fixed. Results of the flattening on different geometries are shown in Figure 7.

6. Application

A possible application for the flattening is result analysis in CFD workflows. We, therefore, integrated our method into a segmentation and simulation pipeline that investigates blood flow in carotid arteries. The overarching goal of this line of research is to better understand the pathophysiology of vascular diseases, e.g., how the arterial wall is affected by blood flow in the vicinity of stenoses.

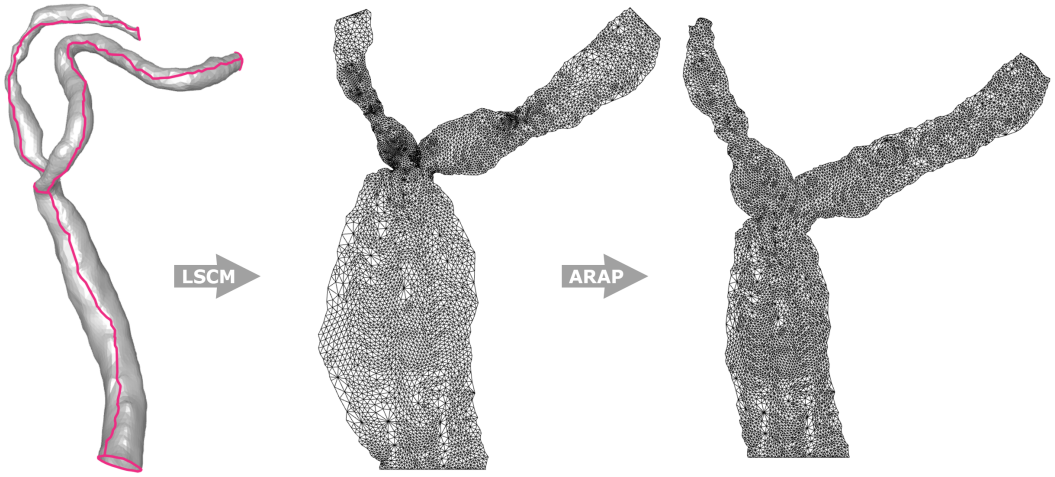


Figure 6: *Stages of the flattening. The cut mesh is first flattened to a predetermined layout using LSCM. Then, ARAP is applied to retain the original proportions.*

These insights could reveal why stenoses often first develop in regions like the carotid bifurcation or siphon and lead to additional markers for clinical decision-making.

The input of the simulation is a model of the carotid inner and outer wall. The data is acquired by solving a fluid-structure-interaction problem. The underlying simulation consists of the incompressible Navier-Stokes equations for a Newtonian fluid and its interaction with a linear elastic vessel wall modeled as a 3D geometry with thickness. The fluid equations are equipped with a velocity function at the inlet boundary and a "do-nothing" boundary condition for the normal stress at the outlets. The vessel wall is considered to behave as a linear elastic material with its outer wall moving freely. The continuity of fluid and solid velocities and the continuity of forces are maintained at the fluid-structure-interface.

Artificial truncation of the vessel results in surfaces perpendicular to the blood flow which are kept stationary throughout the simulation. A density of $\rho_{blood} = 10^3$ kg/m^3 and constant blood viscosity of $\mu = 0.00345$ Pa·s are chosen. The vessel wall has a Young's modulus between $E = 0.25$ - 0.75 MPa, a Poisson's ratio of $\nu = 0.17$ and a density of $\rho_{vessel} = 1.1 \cdot 10^3$ kg/m^3. The cyclic inflow profile of one heart cycle (0.9 s) has peak systolic and diastolic velocities of 0.59 m/s at 0.2 s and 0.29 m/s at 0.55 s respectively. For insights into pathophysiological risk parameters we evaluate the wall shear stress:

$$WSS := \sqrt{\sum_{j=1}^{2} \left((\mathbf{T} \cdot \mathbf{n}) \cdot \tau^{(j)} \right)^2}$$

and the normal wall stress on the fluid-structure interface:

$$NWS := (\mathbf{T} \cdot \mathbf{n}) \cdot \mathbf{n}$$

with the fluid Cauchy stress tensor $\mathbf{T}$, the unit outward normal $\mathbf{n}$, and the unit tangential vectors $\tau^{(j)}$. Parameters that are defined on the whole wall domain, such as displacement, stresses and strains, can either be evaluated on the fluid-structure interface or on the outer vessel wall, since they stay approximately constant in radial directions. The CFD simulation was solved with the FEM based PDE-toolbox Comsol Multiphysics. With around 60k elements and an average element to volume ratio of 10^{-4} the simulation over two cardiac cycles takes approximately 43 hours on two Intel(R) Xeon(R) Gold 5115 CPUs at 2.40 GHz and 256 GB DDR4 RAM.

The simulation results in temporally resolved scalar fields and a displacement vector defined on each vertex. For the output analysis, we use a discretization increment of 0.02 s over the span of 1.8 s. These are effectively 90 time steps that contain two heart cycles. Exploring this spatio-temporal domain is a non-trivial task, which is why we developed a specialized tool for the visual analysis of the result data. As in Figure 1, we show the 3D wall geometry on which any computed scalar field can be color-mapped. The temporal dimension can be explored through animation or selection of time points, where we smoothly interpolate the data between time steps. If wall displacement was computed it can be animated in the 3D view. We provide a GUI to apply the cutting and flattening to the observed model. The method can be seamlessly integrated into the analysis workflow. Cuts automatically face the viewer and the mesh is flattened on demand. The animated colormap is synchronized automatically between the 3D and 2D domains. We found the 2D view especially useful during temporal exploration (see Figure 8), as the whole domain is constantly visible. No interaction is necessary as compared to the 3D rendering, where only ever one side of the model can be shown and self-occlusions of geometry may occur.

6.1. Implementation

The cutting and flattening are implemented in Python, where performance-critical tasks are executed on C++ libraries. The mesh processing largely relies on the implementations in libigl [JP*19]. We use a heap queue for Dijkstra's algorithm to achieve logarithmic complexity. The user can interface with the cutting and flattening through a GUI that also allows exploration of surface fields. All common mesh formats supported by libigl can be loaded. CFD data

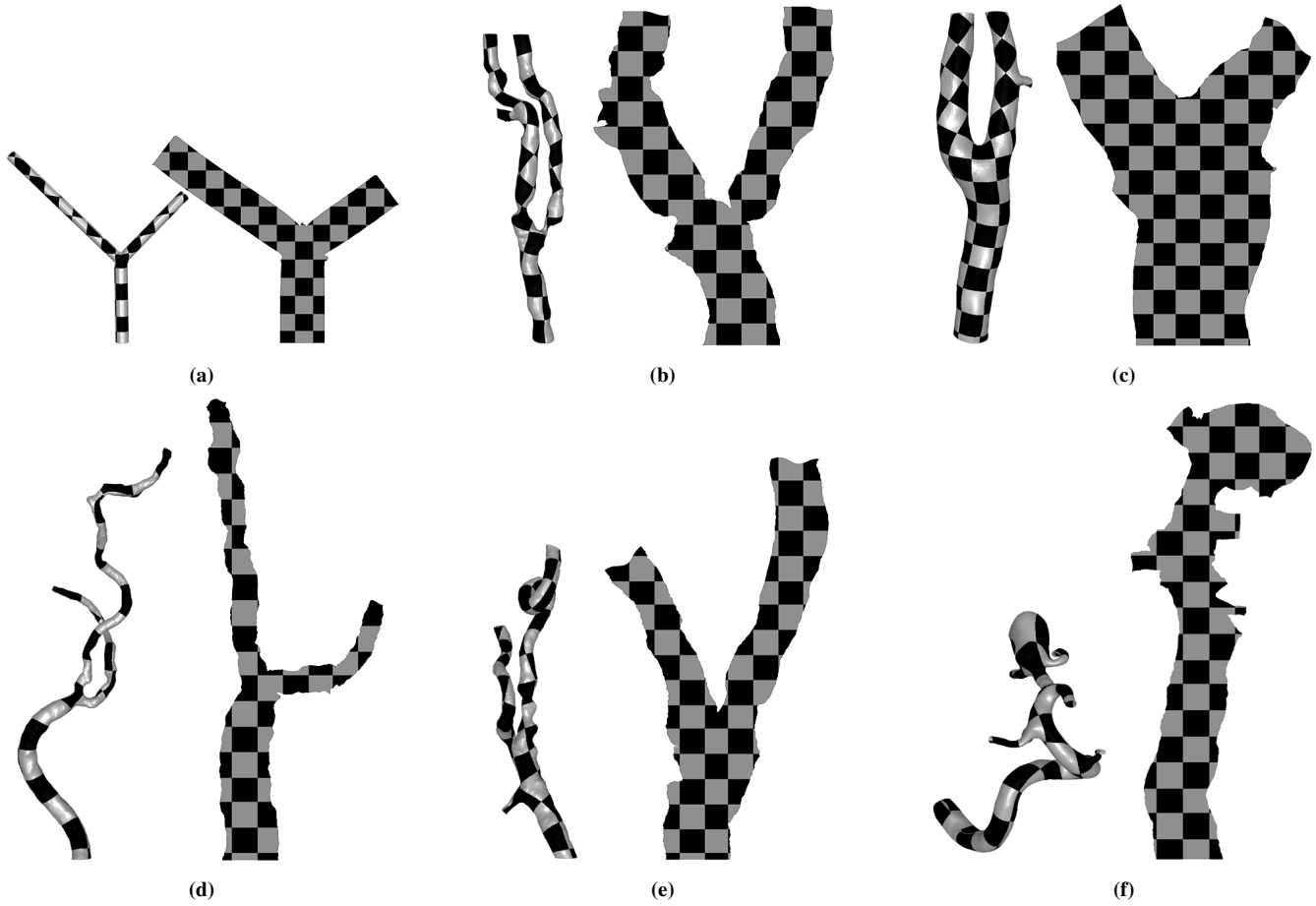


Figure 7: *Our flattening applied to different geometries. Overall, the area is visibly preserved. We keep the general layout of the vessel trees but complex shapes are straightened, yielding a better overview. (a) Phantom data set of a bifurcation. (b)-(e) Human carotid arteries. (f) Cerebral artery with aneurysm.*

is optional and supplied through a table. The application integrates a PyQt GUI with OpenGL 3D rendering. If CFD data is available, we use a texture to pass the surface deformation (three components) and the selected scalar field (one component) to the GPU. The texture contains one entry for each vertex and time step, allowing us to interpolate the temporal dimension at interactive speed.

7. Results and Discussion

We tested our system with carotid vessel trees segmented from five patients. The models we used vary in length, morphology, degree of stenosis, and mesh resolution to cover a broad range of scenarios. Overall, we found the surface area to be well preserved in the 2D mappings (see Figure 7), which is to be expected due to the ARAP optimization. This makes it possible to accurately assess the distribution of scalar fields on the surface. Additionally, the area preservation along with the natural vessel cuts lead to an undistorted visualization of the vessel circumference. This shows where narrow parts of the vessel are located. For instance, in Figure 7c the two stenoses close to the carotid bifurcation are visible in the flattened mesh. We also maintain the general shape of the vessel tree and the orientation of branch locations. This helps mentally mapping the flattened surface to the original 3D structure. Complex bends and twists are straightened by the layout constraints we impose (cf. Figure 5). This filters out unimportant geometrical information and provides a simpler view of the data.

Performance. All tests of the cutting and flattening were executed on an AMD Ryzen 7 5800X processor with 32 GB of RAM. See Table 1 for the impact of different steps and model sizes on the overall performance. For typical use cases of meshes with 10k to 50k triangles the cutting and surface parameterization executes in 1 to 12 s. Therefore, we found the flattening can be easily integrated

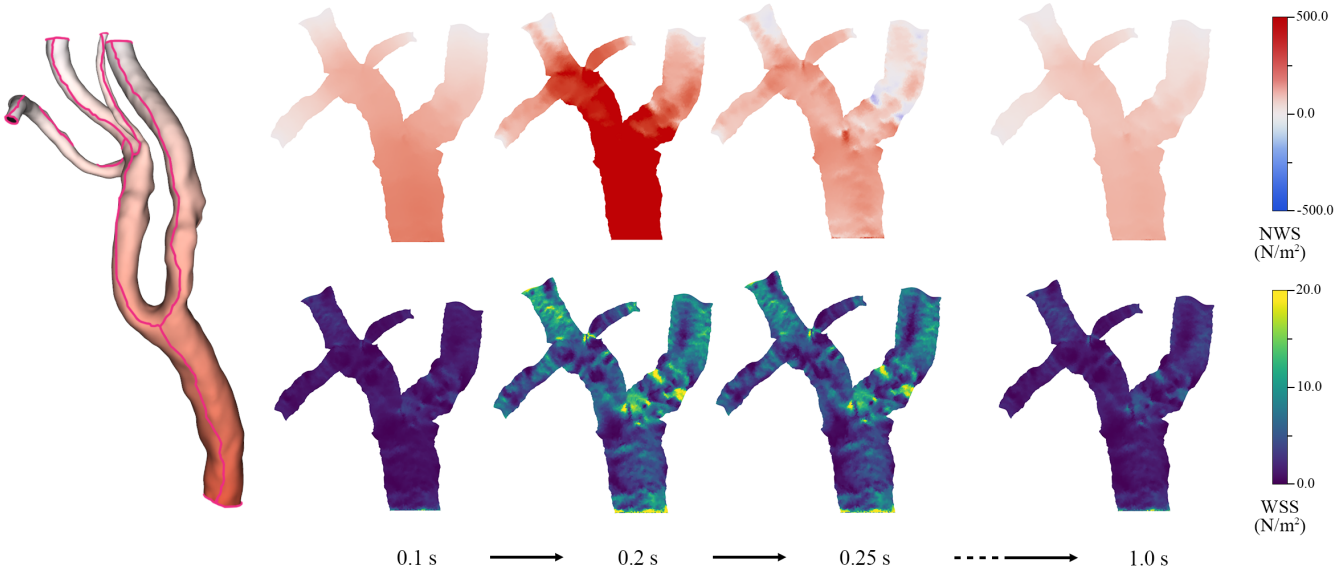


Figure 8: *The flattened geometry can be used to map parameters, like normal wall stress (top) or wall shear stress (bottom). This is particularly beneficial for analysis of the temporal evolution as the whole surface can be assessed at once.*

Table 1: *Timings in seconds of each cutting/flattening step for meshes of different size. $|\mathcal{F}|$: Number of triangles. $\mathcal{Z}$: Computing and aligning curvature. GC: Graph conversion. $\mathcal{V}_C$: Finding cutting paths with Dijkstra's algorithm. AC: Applying the cuts to the mesh topology. L: Layout initialization. F: Flattening stages LSCM and ARAP.*

$\lvert\mathcal{F}\rvert$	$\mathcal{Z}$	GC	$\mathcal{V}_C$	AC	L	F	**Total**
4k	0.1	0.1	0.1	0.1	<0.1	<0.1	**0.5**
13k	0.3	0.4	0.3	0.7	<0.1	0.1	**1.9**
13k	0.3	0.4	0.4	0.4	<0.1	0.1	**1.7**
19k	0.5	0.6	0.5	1.1	<0.1	0.1	**2.8**
20k	0.4	0.7	0.6	1.7	0.1	0.2	**3.6**
24k	0.8	0.8	1.1	2.4	0.1	0.2	**5.3**
28k	0.6	0.9	1.2	2.9	0.1	0.3	**5.9**
44k	1.2	1.4	2.3	5.6	0.4	0.3	**11.2**
300k	16.2	9.1	38.1	109.0	1.0	2.3	**175.8**

into an analysis workflow. The user only needs to set the frontal direction before initiating the cutting, which determines the starting location of the cut on the mesh. The parameter is set automatically by positioning the camera, making this interaction efficient and intuitive. Currently, the slowest step is the application of found cuts to the mesh topology. This bottleneck is due to the indexed face-set data structure we use. Performance could be improved by a integrating a mesh data structure that enables more efficient cutting.

Generalizability. Our application demonstration shows the potential of the flattening for the analysis of carotid wall parameters. Our algorithm is not specific to certain geometries but can be used for a variety of domains, as well as large and small sections of the

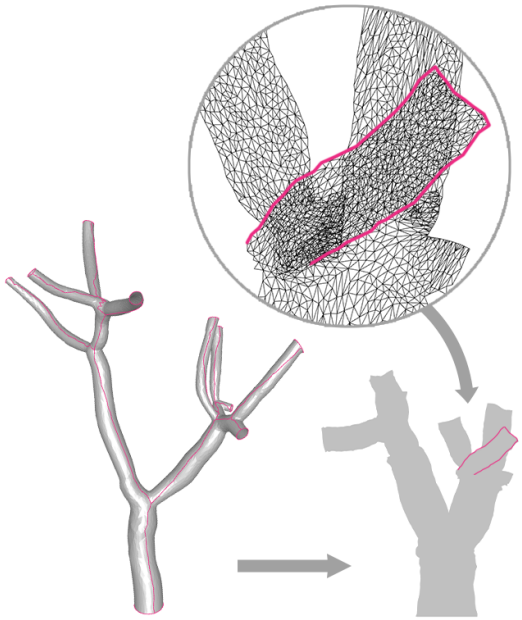


Figure 9: *The flattening does not resolve self-intersections of the layout. This can be problematic for trees with many branches.*

carotids (see Figure 7). The inlet and outlets are automatically detected. Also, the branch layout does not affect the result, i.e., our method is robust against bends or twists of the vessel. The flattening with attribute mapping would also be equivalently applicable for measured instead of simulated wall properties, for instance, plaque occurrence or wall thickness. Furthermore, we demonstrate

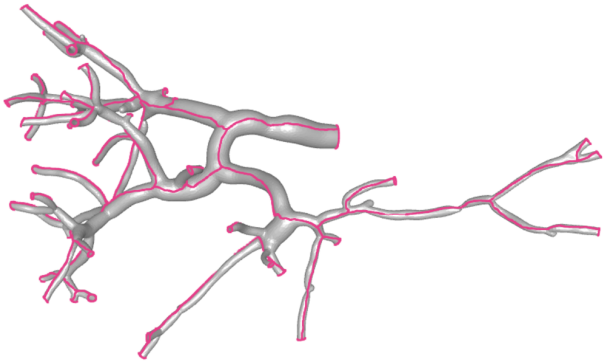


Figure 10: *The cutting performs well even on complex structures like this arterial tree segmented from a liver.*

that the method can be applied to other vessel trees, not just the carotids. We show examples of cerebral and aortic vessel trees with aneurysms (cf. Figures 4, 7), which pose similar problems in CFD and medical research. Last but not least, the cutting is independent of the flattening stage. As a result, natural vessel cuts could be used in other scenarios or with a completely different flattening approach.

Limitations. LSCM and ARAP ensure that no triangle flips occur and the surface is parameterized as a single patch. However, self-intersections may still appear in the parameter domain. In some models, we could observe this happening close to branch locations. These overlaps could only be solved by introducing area deformation. Also, they only affect small portions of the parameter domain. Larger overlaps can ensue from more complex vessel trees with numerous branches and sub-branches. In Figure 9 the flattening produces noticeable artifacts as the ARAP optimization relaxes the mesh such that it intersects itself. Our layout initialization (see Figure 5) does not account for possible overlaps of segments. Therefore, we would advise against using the flattening method for overly complex vessel trees. The cutting, however, is not affected by the number of branches and can be applied to any tree (see Figure 10).

8. Conclusion

We presented a fully automatic technique for cutting and flattening vessel tree geometries. We introduced natural vessel cuts that remain on one side of cylindrical vessel segments and face the observer when possible. Furthermore, we demonstrated a use case for the analysis of fluid-wall interaction simulations in the human carotids. The application is not limited to the carotids but can be generalized to similar tree-like structures. In contrast to other approaches, we require only the surface mesh as an input and create a boundary-free parameterization that is quasi-isometric and retains the original vessel shape. The cutting is robust and will always return a valid disk-cut given a tree-like geometry, where the cut connections will be at branch locations. The flattening always results in a single connected patch. We presented an automatic way to determine a layout for the flattened geometry that can be used for the carotids or vessel trees with similar complexity.

In the future, the layout initialization could be improved with additional constraints that prevent overlaps of segments. Further ideas include leveraging the flattened geometry for more advanced visual exploration and interaction techniques. Camera control, as well as brushing and linking of interesting regions, could be facilitated by the flattened geometry. Due to its visual simplicity, the 2D depiction could also be used to create comparative views of data set collections. Such overviews would be advantageous when comparing patient cohorts or CFD parameter sets.

Acknowledgements

This work was partially funded by the BMBF Joint Project 05M20SJA-MLgSA and the Carl-Zeiss foundation.

References

[AS04] ANTIGA, LUCA and STEINMAN, DAVID A. "Robust and objective decomposition and mapping of bifurcating vessels". *IEEE Transactions on Medical Imaging* 23.6 (2004), 704–713. DOI: 10.1109/TMI.2004.826946 2, 3.

[BGP*11] BORKIN, MICHELLE, GAJOS, KRZYSZTOF, PETERS, AMANDA, et al. "Evaluation of artery visualizations for heart disease diagnosis". *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), 2479–2488. DOI: 10.1109/TVCG.2011.192 2, 3.

[CCC16] CHIU, BERNARD, CHEN, WEIFU, and CHENG, JIEYU. "Concise biomarker for spatial–temporal change in three-dimensional ultrasound measurement of carotid vessel wall and plaque thickness based on a graph-based random walk framework: Towards sensitive evaluation of response to therapy". *Computers in Biology and Medicine* 79 (2016), 149–162. DOI: 10.1016/j.compbiomed.2016.10.015 3.

[CCR20] CHOI, GARY PT, CHIU, BERNARD, and RYCROFT, CHRIS H. "Area-preserving mapping of 3D carotid ultrasound images using density-equalizing reference map". *IEEE Transactions on Biomedical Engineering* 67.9 (2020), 2507–2517. DOI: 10.1109/TBME.2019.2963783 2, 3.

[CES*08] CHIU, BERNARD, EGGER, MICAELA, SPENCE, DAVID J, et al. "Area-preserving flattening maps of 3D ultrasound carotid arteries images". *Medical Image Analysis* 12.6 (2008), 676–688. DOI: 10.1016/j.media.2008.04.002 2, 3.

[CGRH16] CHOUDHRY, FAROOQ A, GRANTHAM, JOHN T, RAI, ANSAAR T, and HOGG, JEFFERY P. "Vascular geometry of the extracranial carotid arteries: an analysis of length, diameter, and tortuosity". *Journal of Neurointerventional Surgery* 8.5 (2016), 536–540. DOI: 10.1136/neurintsurg-2015-011671 3.

[CL13] CABALLERO, ANDRES D and LAÍN, SANTIAGO. "A review on computational fluid dynamics modelling in human thoracic aorta". *Cardiovascular Engineering and Technology* 4.2 (2013), 103–130. DOI: 10.1007/s13239-013-0146-6 2.

[CLC13] CHIU, BERNARD, LI, BING, and CHOW, TOMMY WS. "Novel 3D ultrasound image-based biomarkers based on a feature selection from a 2D standardized vessel wall thickness map: a tool for sensitive assessment of therapies for carotid atherosclerosis". *Physics in Medicine & Biology* 58.17 (2013), 5959–5982. DOI: 10.1088/0031-9155/58/17/5959 3.

[CUS*17] CHENG, JIEYU, UKWATTA, ERANGA, SHAVAKH, SHADI, et al. "Sensitive three-dimensional ultrasound assessment of carotid atherosclerosis by weighted average of local vessel wall and plaque thickness change". *Medical Physics* 44.10 (2017), 5280–5292. DOI: doi.org/10.1002/mp.12507 3.

[CWW13] CRANE, KEENAN, WEISCHEDEL, CLARISSE, and WARDETZKY, MAX. "Geodesics in heat: a new approach to computing distance based on heat flow". *ACM Transactions on Graphics* 32.5 (2013), 1–11. DOI: 10.1145/2516971.2516977 5.

[DHS*13] DIEPENBROCK, STEFAN, HERMANN, SVEN, SCHÄFERS, MICHAEL, et al. "Comparative visualization of tracer uptake in in vivo small animal PET/CT imaging of the carotid arteries". *Computer Graphics Forum*. Vol. 32. 3pt2. 2013, 241–250. DOI: 10.1111/cgf.12111 2.

[Dij59] DIJKSTRA, EDSGER W. "A note on two problems in connexion with graphs". *Numerische Mathematik* 1.1 (1959), 269–271. DOI: 10.1007/BF01386390 5.

[EEL*19] EULZER, PEPE, ENGELHARDT, SANDY, LICHTENBERG, NILS, et al. "Temporal views of flattened mitral valve geometries". *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), 971–980. DOI: 10.1109/TVCG.2019.2934337 2.

[EMKL21] EULZER, PEPE, MEUSCHKE, MONIQUE, KLINGNER, CARSTEN M, and LAWONN, KAI. "Visualizing carotid blood flow simulations for stroke prevention". *Computer Graphics Forum*. Vol. 40. 2021, 435–446. DOI: 10.1111/cgf.14319 2, 6.

[GBD16] GBD 2015 MORTALITY AND CAUSES OF DEATH COLLABORATORS. "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015". *The Lancet* 388.10053 (2016), 1459–1544. ISSN: 0140-6736. DOI: 10.1016/S0140-6736(16)31012-1 2.

[GGH02] GU, XIANFENG, GORTLER, STEVEN J, and HOPPE, HUGUES. "Geometry images". *Proceedings of the 29th annual conference on computer graphics and interactive techniques*. 2002, 355–361. DOI: 10.1145/566570.566589 4.

[GSK*12] GOUBERGRITS, LEONID, SCHALLER, JENS, KERTZSCHER, ULRICH, et al. "Statistical wall shear stress maps of ruptured and unruptured middle cerebral artery aneurysms". *Journal of the Royal Society Interface* 9.69 (2012), 677–688. DOI: 10.1098/rsif.2011.0490 2, 3.

[JLM*17] JOSEPH, PHILIP, LEONG, DARRYL, MCKEE, MARTIN, et al. "Reducing the global burden of cardiovascular disease, part 1: the epidemiology and risk factors". *Circulation Research* 121.6 (2017), 677–694. DOI: 10.1161/CIRCRESAHA.117.308903 2.

[JP*19] JACOBSON, ALEC, PANOZZO, DANIELE, et al. *libigl: A simple C++ geometry processing library*. https://libigl.github.io/. 2019 7.

[KFW*02] KANITSAR, A., FLEISCHMANN, D., WEGENKITTL, R., et al. "CPR - curved planar reformation". *IEEE Visualization*. 2002, 37–44. DOI: 10.1109/VISUAL.2002.1183754 2.

[KMM*18] KREISER, JULIAN, MEUSCHKE, MONIQUE, MISTELBAUER, GABRIEL, et al. "A survey of flattening-based medical visualization techniques". *Computer Graphics Forum*. Vol. 37. 2018, 597–624. DOI: 10.1111/cgf.13445 2.

[KNP11] KÄLBERER, FELIX, NIESER, MATTHIAS, and POLTHIER, KONRAD. "Stripe parameterization of tubular surfaces". *Topological Methods in Data Analysis and Visualization*. 2011, 13–26. DOI: 10.1007/978-3-642-15014-2_2 3.

[LER*20] LICHTENBERG, NILS, EULZER, PEPE, ROMANO, GABRIELE, et al. "Mitral valve flattening and parameter mapping for patient-specific valve diagnosis". *International Journal of Computer Assisted Radiology and Surgery* (2020), 617–627. DOI: 10.1007/s11548-019-02114-w 2.

[LL20] LICHTENBERG, NILS and LAWONN, KAI. "Parameterization, Feature Extraction and Binary Encoding for the Visualization of Tree-Like Structures". *Computer Graphics Forum*. Vol. 39. 1. 2020, 497–510. DOI: 10.1111/cgf.13888 3, 5.

[LPRM02] LÉVY, BRUNO, PETITJEAN, SYLVAIN, RAY, NICOLAS, and MAILLOT, JÉROME. "Least squares conformal maps for automatic texture atlas generation". *ACM Transactions on Graphics* 21.3 (2002), 362–371. DOI: 10.1145/566654.566590 6.

[LZX*08] LIU, LIGANG, ZHANG, LEI, XU, YIN, et al. "A local/global approach to mesh parameterization". *Computer Graphics Forum*. Vol. 27. 5. 2008, 1495–1504. DOI: 10.1111/j.1467-8659.2008.01290.x 6.

[MK15] MARINO, JOSEPH and KAUFMAN, ARIE. "Planar visualization of treelike structures". *IEEE Transactions on Visualization and Computer Graphics* 22.1 (2015), 906–915. DOI: 10.1109/TVCG.2015.2467413 2–4.

[MMV*13] MISTELBAUER, GABRIEL, MORAR, ANCA, VARCHOLA, ANDREJ, et al. "Vessel visualization using curvicircular feature aggregation". *Computer Graphics Forum*. Vol. 32. 3pt2. 2013, 231–240. DOI: 10.1111/cgf.12110 2.

[MPG*17] MARTINKE, HANNES, PETRY, CHRISTIAN, GROSSKOPF, STEFAN, et al. "Bone Fracture and Lesion Assessment using Shape-Adaptive Unfolding." *Eurographics Workshop on Visual Computing for Biology and Medicine*. 2017, 149–158 2.

[MVB*12] MISTELBAUER, GABRIEL, VARCHOLA, ANDREJ, BOUZARI, HAMED, et al. "Centerline reformations of complex vascular structures". *IEEE Pacific Visualization Symposium*. 2012, 233–240. DOI: 10.1109/PacificVis.2012.6183596 2.

[MVB*16] MEUSCHKE, MONIQUE, VOSS, SAMUEL, BEUING, OLIVER, et al. "Combined visualization of vessel deformation and hemodynamics in cerebral aneurysms". *IEEE Transactions on Visualization and Computer Graphics* 23.1 (2016), 761–770. DOI: 10.1109/TVCG.2016.2598795 3.

[MZGK11] MARINO, JOSEPH, ZENG, WEI, GU, XIANFENG, and KAUFMAN, ARIE. "Context preserving maps of tubular structures". *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), 1997–2004. DOI: 10.1109/TVCG.2011.182 2.

[NGB*09] NEUGEBAUER, MATHIAS, GASTEIGER, ROCCO, BEUING, OLIVER, et al. "Map displays for the analysis of scalar data on cerebral aneurysm surfaces". *Computer Graphics Forum*. Vol. 28. 3. 2009, 895–902. DOI: 10.1111/j.1467-8659.2009.01459.x 2.

[OMN*19] OELTZE-JAFRA, STEFFEN, MEUSCHKE, MONIQUE, NEUGEBAUER, MATHIAS, et al. "Generation and visual exploration of medical flow data: Survey, research trends and future challenges". *Computer Graphics Forum*. Vol. 38. 2019, 87–125. DOI: 10.1111/cgf.13394 2.

[PPR10] PANOZZO, DANIELE, PUPPO, ENRICO, and ROCCA, LUIGI. "Efficient multi-scale curvature and crease estimation". *Proceedings of Computer Graphics, Computer Vision and Mathematics* 1.6 (2010) 5.

[PSY*19] PANDEY, ADITEYA, SHUKLA, HARSH, YOUNG, GEOFFREY S, et al. "Cerebrovis: designing an abstract yet spatially contextualized cerebral artery network visualization". *IEEE Transactions on Visualization and Computer Graphics* 26.1 (2019), 938–948. DOI: 10.1109/TVCG.2019.2934402 2.

[SA07] SORKINE, OLGA and ALEXA, MARC. "As-rigid-as-possible surface modeling". *Symposium on Geometry processing*. Vol. 4. 2007, 109–116 6.

[SH18] SZAJER, JEREMY and HO-SHON, KEVIN. "A comparison of 4D flow MRI-derived wall shear stress with computational fluid dynamics methods for intracranial aneurysms and carotid bifurcations — A review". *Magnetic Resonance Imaging* 48 (2018), 62–69. ISSN: 0730-725X. DOI: 10.1016/j.mri.2017.12.005 2.

[THQ*16] TAO, JUN, HUANG, XIAOKE, QIU, FENG, et al. "VesselMap: A web interface to explore multivariate vascular data". *Computers & Graphics* 59 (2016), 79–92. DOI: 10.1016/j.cag.2016.05.024 2.

[ZHT05] ZHU, LEI, HAKER, S., and TANNENBAUM, A. "Flattening maps for the visualization of multibranched vessels". *IEEE Transactions on Medical Imaging* 24.2 (2005), 191–198. DOI: 10.1109/TMI.2004.839368 3.