DOI: 10.1111/cgf.15135

Eurographics Symposium on Geometry Processing 2024
R. Hu and S. Lefebvre
(Guest Editors)

COMPUTER GRAPHICS *forum*
Volume 43 (2024), Number 5

Distance-Based Smoothing of Curves on Surface Meshes

M. Pawellek[1] and C. Rössl[2] and K. Lawonn[1]

[1]University of Jena, Faculty of Mathematics and Computer Science, Germany
[2]University of Magdeburg, Faculty of Computer Science, Germany

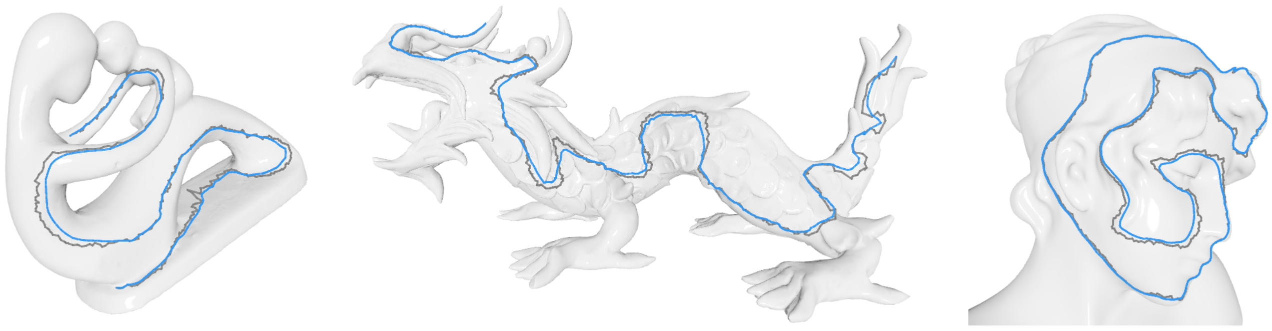


Figure 1: *Surface meshes with initial (black) and smoothed (blue) surface curves generated by our algorithm.*

Abstract

The smoothing of surface curves is an essential tool in mesh processing, important to applications that require segmenting and cutting surfaces such as surgical planning. Surface curves are typically designed by professionals to match certain surface features. For this reason, the smoothed curves should be close to the original and easily adjustable by the user in interactive tools. Previous methods achieve this desired behavior, e.g., by utilizing energy-minimizing splines or generalizations of Bézier splines, which require a significant number of control points and may not provide interactive frame rates or numerical stability. This paper presents a new algorithm for robust smoothing of discrete surface curves on triangular surface meshes. By using a scalar penalty potential as the fourth coordinate, the given surface mesh is embedded into the 4D Euclidean space. Our method is based on finding geodesics in this lifted surface, which are then projected back onto the original 3D surface. The benefits of this approach include guaranteed convergence and good approximation of the initial curve. We propose a family of penalty potentials with one single parameter for adjusting the trade-off between smoothness and similarity. The implementation of our method is straightforward as we rely on existing methods for computing geodesics and penalty fields. We evaluate our implementation and confirm its robustness and efficiency.

CCS Concepts
• ***Computing methodologies*** → *Mesh geometry models;*

1. Introduction

Curves on surface meshes are a basic building block for mesh processing and segmentation (see, e.g., JI et al. [JLCW06], KAPLANSKY and TAL [KT09], and LIVESU [Liv18]). Medical applications, such as the resection of liver tumors [AR20], osteotomy planning [ZGSZ03], or machine learning-assisted segmentation of CT data [PYK*20], are particular use cases. Other applications include illustrative visualization of medical data sets [LVPI18] or engineering [BLVD11].

In these application scenarios, surface curves are chosen manually or automatically. These initial curves typically show noise, e.g., small oscillations, which may originate from the finite precision of the underlying surface mesh and the specific curve painting strategies [Liv18]. These artifacts are not only perceivable by the human eye, but they are generally not negligible for further processing like segmentation and/or surface cutting [LGRP14]. Consequently, a smoothing process must be applied to suppress noise, i.e., to reduce the total (geodesic) curvature. This process has often competing objectives: the result is expected to be smooth, however, the smoothed

© 2024 The Authors. Computer Graphics Forum published by Eurographics - The European Association for Computer Graphics and John Wiley & Sons Ltd.
This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

delivered by EUROGRAPHICS DIGITAL LIBRARY
www.eg.org diglib.eg.org

curve should not deviate too much from the initial curve. The second requirement is crucial in many cases: For medical surface-cutting applications, for example, the shape of the initial curve is defined by domain experts, such as physicians or bio-engineers, and most likely indicates relevant anatomical landmarks or surface regions. Therefore, the smoothing must preserve a certain degree of similarity to its original so that no essential information is lost.

At first glance, smoothing surface curves may appear as a simple problem with straightforward solutions. However, the restriction of the curve to the curved surface makes this a difficult problem, which is often underestimated. This is even more true, as many applications require interactive control and robust results even for poor surface tessellations. Previous solutions to this problem include the use of energy-minimizing splines [HP04], generalizations of Bézier curves in Euclidean space to curves on surfaces embedded in 3D [MCV07; MNPP23], the iterative reduction of the geodesic curvature [LGRP14], or the application of Laplacian smoothing to the heat gradient [Liv18]. All of these methods show advantages and disadvantages. Numerically robust smoothing may be difficult to achieve at interactive frame rates due to computationally expensive operations, or it requires a great deal of user interaction to achieve a desired result [LGRP14].

In this paper, we present a novel algorithm for efficient and robust smoothing of simple, discrete surface curves that are given on triangular surface meshes. In the absence of a precise definition of *smoothing* in the literature [LGRP14; Liv18], we adopt the understanding that smoothing is a *denoising* process: Given an imperfect, noisy surface curve, we find a smoother, "straighter" curve in the same manifold that is close to the initial curve. Thus, the smoothing of curves is seen as reducing the local geodesic curvature of the initially given curve similar to LAWONN et al. [LGRP14]. In this sense, geodesics are the smoothest possible curves since they show zero geodesic curvature. Although possible, our method does not aim at the design of fair surface curves in the sense of finding the curve with the simplest shape subject to boundary conditions.

Our approach lifts the original surface and the surface curve into a higher dimensional space and then constructs a potential that penalizes the deviation from the initially given curve. The lifted surface is a topological 2-manifold embedded in the 4D Euclidean space. The lifted curve, now embedded in the lifted surface, is then *straightened* to become a (locally shortest) geodesic in the lifted surface. The purpose of lifting is to change the metric, specifically the measurement of the arc length, to take into account the proximity to the original curve. The generated geodesic belongs to the same isotopy class as the initial curve and is a smooth surface curve by construction: straight, with zero geodesic curvature in a smooth setting or locally shortest on discrete surfaces. The projection of this geodesic from the 4D surface back onto the 3D surface yields a smooth curve that is close to the initial curve. Our method is conceptually simple but offers significant advantages. To summarize, the main contributions are

- Concise non-iterative algorithm with guaranteed convergence.
- Single-parameter to adjust closeness versus smoothness.
- Straightforward implementation with interactive performance.

2. Related Work

Our method depends on the tracing of discrete geodesics and the construction of (geodesic) distance fields on surface meshes. The tracing of geodesics manifests either by shortening an open curve with fixed start and end points or by shrinking a closed loop. Most existing local methods for tracing geodesics employ an optimization approach and are capable of handling both of these cases [CLPQ20]. For a detailed overview, we refer to the surveys BOSE et al. [BMSW11], PATANÉ [Pat16], and CRANE et al. [CLPQ20]. Roughly speaking, algorithms for computing geodesic distances or shortest geodesic paths on meshes either establish local isometric parameterizations for each path by locally "flattening" sequences of triangles, or they rely on solving a partial differential equation like the heat or wave equation.

The most intuitive approach to the design of discrete surface curves by smoothing a given input curve probably consists of using subdivision techniques on polygons. This was first proposed by MORERA et al. [MVC08]. The resulting curves, however, may include line segments that are not included in the surface and, thus, may violate the manifold constraint [JSW*19]. Furthermore, the algorithm does not offer any parameters to control the trade-off between similarity and smoothness. Besides, the generation of a smooth curve from "fixed" discrete control points is not a smoothing technique in the sense of denoising.

In geometry processing, smooth (planar or 3D) curves are often represented as splines, i.e., piecewise smooth curves. This concept has been extended to surface curves. HA et al. [HPY21] use the conventional Hermite interpolation method by exchanging primitive operations between points and vectors by geodetic queries such that generated curves always fulfill the manifold constraint. HOFER and POTTMANN [HP04] and POTTMANN and HOFER [PH05] determine splines in general manifolds by modeling the design of smooth curves as an optimization problem. They use a variational approach and obtain smooth curves as minimizers of a quadratic energy. Their approaches are not only applicable to curves on surface meshes but can be used for a much broader variety of cases, including, e.g., the design of rigid body motions. Alas, the method's generality leads to computationally expensive operations and compromises its performance [JSW*19]. JIN et al. [JSW*19] address this issue by using a shell space approach, resulting in improved performance and numerically robust results. Still, the major drawback of variational methods is that enforcing closeness to the initial curve involves the manual handling of many control points for the user [LGRP14]. In general, these approaches are not expected to provide interactive frame rates for high-resolution surface meshes [MNPP23].

An alternative to energy-minimizing surface splines is the generalization of Bézier curves or piecewise Bézier curves, i.e., splines, to surface curves as done by MORERA et al. [MCV07], XU et al. [XJZ*23], and MANCINELLI et al. [MNPP23]. The idea is to lift 2D Bézier curves in Euclidean space to geodesic Bézier splines located in the surface with the initial curve samples used as control points. By introducing a scalar potential inside a shell space around the surface mesh, even feature-aware curves can be designed [XJZ*23]. Approaches based on Bézier curves seem to be superior to other spline-based methods as they offer real-time performance even for high-resolution meshes [MNPP23]. Nevertheless, they have in common with variational methods that only the use of many control points can ensure the closeness of the resulting curves to the original. Also, a similar remark as for subdivision curves applies: the evaluation of a smooth surface curve for given control points is not a general smoothing technique. In summary, these approaches do not fit our smoothing scenario of suppressing noise. Their strengths consist in the design of smooth surface curves in particular for vector graphics on surface meshes [MNPP23].

Another well-known approach uses surface features for automatic mesh segmentation and cutting (see, e.g., SHAMIR [Sha08] for a general survey). To classify surface features, LAI et al. [LZH*07] use a feature-sensitive curve smoothing, which yields smooth boundaries for mesh features. JUNG and KIM [JK04] and BISCHOFF et al. [BWK05] generalize so-called snakes or active contours well-known from image processing on 2-manifolds to represent curves on surfaces that "snap" to mesh features starting from an initial curve. Snakes are closed curves that evolve over time, adapting to features as they move. The iterative process is driven by minimizing internal and external forces that are based on curvature, length, and distance to features. While solving a related problem, feature-based methods are different to general smoothing, because the main focus is on matching the given features whereas smoothness acts as a "regularization" when features are locally absent.

A different approach is presented by LAWONN et al. [LGRP14]: Their method adapts a Laplacian smoothing where the movement of the curve's vertices is restricted to the edges of the surface mesh and vertices are allowed to split and merge, hence expanding or shrinking the number of curve segments. A restriction by a distance envelope ensures that the result remains close to the initial curve. This construction guarantees restriction to surface curves and the authors show convergence of the iterative algorithm. Additionally, a global parameter can be used to adjust the trade-off between similarity and smoothness. However, the number of iterations and the performance vary significantly with the use cases, and the method may be too slow for interactive use.

Moreover, LIVESU [Liv18] introduced an elegant relaxation scheme akin to the heat method [CWW13]. This approach effectively smooths the boundary between two specified components. By applying Laplacian smoothing to the gradient of the heat field generated by the initial curve, the resulting curve is represented by a level set of the smoothed gradient field's potential. Notably, the algorithm is not restricted to curves in 2-manifolds but can be applied for general n-dimensional hyper-surfaces. The algorithm is primarily employed in the realm of automatic mesh segmentation and limited to bi-partitions and, consequently, to closed-edge curves that partition the surface into two distinct subsets and do not self-intersect. For curves in high-resolution surface meshes, this method requires several seconds or even minutes and thus lacks interactivity.

The methods of LAWONN et al. [LGRP14] and LIVESU [Liv18] are closest to our problem scenario. Unfortunately, neither method provides a rigorous definition of smoothing that would allow for a quantitative comparison. Instead, they discuss specific applications to illustrate the concept by example. In our approach, the smoothing of initially provided discrete surface mesh curves is conceptualized as as a reduction in local geodesic curvatures while preserving the manifold constraint. This is similar to LAWONN et al. [LGRP14]. By employing a specific one-parameter family of penalty potentials, our approach facilitates adjusting the balance between similarity and smoothness without the need for any manual control point definition. The construction of these penalty potentials from (geodesic) distance fields ensures that the smoothing serves as intended feature-independent regularization near the original curve. Given an efficient and robust algorithm for the construction of discrete geodesics, our approach consists of finitely many steps, is guaranteed to compute a solution, and shows interactive performance for high-resolution meshes for open and closed curves.

The fundamental approach of our algorithm is to lift the geometry to a higher dimension, thereby transforming a challenging optimization problem with manifold constraints into the more straightforward problem of tracing geodesics. The general concept of lifting is not new, in particular, lifting techniques have been successfully applied in geometry processing: In a series of papers Du et al. [DAZ*20; DKZ*21; DKZ*22] utilize lifting schemes for constructing guaranteed bijective surface parametrizations given nontrivial and nonconvex boundary curves.

3. Algorithm

Our algorithm consists of four conceptual stages that can be purely described in terms of operations on smooth manifolds. The input is given by a surface $\mathcal{S}$ and an initial surface curve γ in $\mathcal{S}$. The output is denoted by λ, which again is a surface curve in $\mathcal{S}$. Mathematically, $\mathcal{S}$ is a smooth 2D Riemannian submanifold embedded in 3D Euclidean space. Both, γ and λ, are simple curves in $\mathcal{S}$ parameterized by arc-length. λ is of class C^1 and γ is assumed to be sufficiently smooth. Furthermore, for a scalar function $\varphi : \mathcal{S} \to \mathbb{R}$, the lifting operator χ_φ is given as

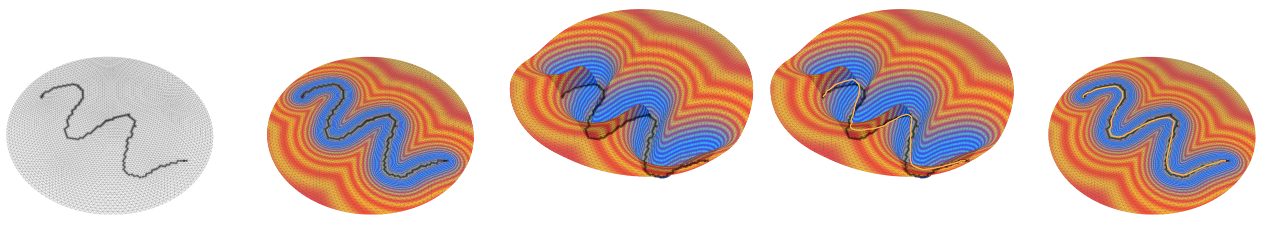
$$\chi_\varphi : \mathcal{S} \to \mathbb{R}^4 \quad \text{with} \quad \chi_\varphi(\mathbf{x}) := (\mathbf{x}, \varphi(\mathbf{x})) .$$

Using this notation, our algorithm can be summarized as follows.

```
Algorithm: Geodetic Smoothing

1. Construct a potential $\varphi\colon \mathcal{S} \to \mathbb{R}$
   that penalizes larger distances to $\gamma$ by higher values.
2. Apply $\chi_\varphi$ to $\mathcal{S}$ and $\gamma$ to get $\mathcal{S}_\varphi := \chi_\varphi(\mathcal{S})$ and $\gamma_\varphi := \chi_\varphi \circ \gamma$.
3. Determine a (locally shortest) geodesic $\lambda_\varphi$ in $\mathcal{S}_\varphi$
   that lies in the isotopy class of $\gamma_\varphi$.
4. Find $\lambda$ such that $\lambda_\varphi = \chi_\varphi \circ \lambda$ holds.
```

Figure 2 demonstrates the stages of our method for a planar surface. In this 2D case, the lifted surface $\mathcal{S}_\varphi$ can be embedded in the 3D Euclidean space for illustration. To provide an intuitive explanation, imagine the potential as a relief of mountains, with the initial curve representing a river flowing through a valley. The river is located at the lowest height potential, which is constant, i.e., the river flows "magically" without gravity. The shape of the mountain relief allows the river to "cut through": Over time, the river naturally straightens, influenced by its current, always following the locally shortest path: it gets smoothed. Hereby, the mountains serve as obstacles, preventing the river from becoming a straight line and deviating too much from its original shape. Thus, the incorporation of a penalty potential serves as a distance-based constraint, while tracing the locally



(a) *2D Input* **(b)** *Step 1: Penalty Potential* **(c)** *Step 2: Lifting to 3D* **(d)** *Step 3: Geodesic in 3D* **(e)** *Step 4: Projection to 2D*

Figure 2: *Application of the algorithm to a planar surface. The stages of our algorithm are shown from left to right by using a surface with an initially jagged surface curve (black). As the surface is planar, the lifted surface can be embedded and visualized in 3D Euclidean space. The penalty potential is color-coded from blue (low) to red (high). The lifted surface is curved due to the penalty potential, here as height coordinate. The geodesic and its projection are shown as orange curves.*

shortest geodesic functions as the smoothing strategy to relax the curve and remove noise. It is straightforward to extend this concept from planar surfaces to curved surfaces in 3D Euclidean space and thus from planar curves to surface curves: introduce the potential as a fourth coordinate to the surface position, effectively lifting the surface into 4D Euclidean space.

In a more formal explanation, the first step of our algorithm constructs the penalty potential φ. This is non-trivial and is discussed in Section 3.3. For the moment, we assume that φ is given and maps points further away from the initial curve to higher scalar values. In the second step, the lifting operator χ_φ utilizes φ as the fourth coordinate for positions. This way, the surface $\mathcal{S}$ and the initial curve γ are lifted into the 4D Euclidean space. Note that the lifted surface $\mathcal{S}_\varphi$ now bends differently in 4D space and exhibits a modified surface metric compared to its 3D counterpart, similar to mountains versus plane in the illustrative example. We exploit this change in the surface metric in step three and determine a geodesic λ_φ in $\mathcal{S}_\varphi$ and γ_φ's isotopy class, that automatically adjusts itself to the shape of γ_φ, either by shrinking a closed loop or shortening an open curve with fixed endpoints. The fourth step is trivial: it only projects λ_φ back into 3D by dropping the fourth coordinate. This yields our result λ. Please note, that this final step does not preserve the geodesic property, i.e., the smoothed surface curve λ is in general not a (locally shortest) geodesic curve in $\mathcal{S}$.

3.1. Existence and Uniqueness

Assume that φ is a smooth function. Then χ_φ is a smooth embedding as $\chi_\varphi\colon \mathcal{S} \to \mathcal{S}_\varphi$ is a homeomorphism with inverse χ_φ^{-1} and its differential $d\chi_\varphi(\mathbf{x})$ is injective for all $\mathbf{x} \in \mathcal{S}$. Consequently, $\mathcal{S}_\varphi$ is a smooth 2D Riemannian submanifold embedded in 4D Euclidean space. Let $\Gamma_\gamma(\mathcal{S})$ be the set of C^1 surface curves in $\mathcal{S}$ that are also part of γ's isotopy class. In the case of γ being open, their endpoints additionally are required to coincide with γ's endpoints. For the lifted initial curve γ_φ, there must exist at least one geodesic $\lambda_\varphi \in \Gamma_{\gamma_\varphi}(\mathcal{S}_\varphi)$. In the absence of boundaries, λ_φ must be smooth [AA81]. Using the inverse χ_φ^{-1} as smooth projection, it follows that $\lambda = \chi_\varphi^{-1} \circ \lambda_\varphi$ must be smooth as well. In the more general setting of manifolds with non-empty boundaries, λ_φ and, consequently, λ are at least of class C^1 [AA81; ABB87].

It is generally observed that λ_φ and λ are not necessarily unique as, for instance, two antipodal points connected by γ on the unit disk could be lifted onto a hemisphere, resulting in an infinite number of possible geodesics that solve the boundary value problem in the lifted surface. Nevertheless, geodesic tracing techniques, such as the *FlipOut* algorithm, are capable of identifying one of these curves, even in the presence of other geodesics in the same isotopy class [CLPQ20]. This is typically achieved by leveraging an iterative optimization approach [MVC05; SC20; MNPP23]. Since the initial guess provided in the form of γ_φ is used as the basis for this optimization, the method is more likely to converge to a closer local minimum.

3.2. Smoothness and Closeness

To demonstrate that the algorithm indeed yields a result representing a relaxed version of the original, similar in shape, we employ the common strategy of reformulating its result as a local solution to an optimization problem. Geodesics in Riemannian manifolds can be characterized as local minima of an energy functional. With that, step three of our algorithm can be expressed as a local solution to the following optimization problem:

$$\underset{\alpha \in \Gamma_{\gamma_\varphi}(\mathcal{S}_\varphi)}{\arg\min}\ E(\alpha) \quad \text{with} \quad E(\alpha) := \frac{1}{2}\int_{\mathcal{D}(\alpha)} \|\dot\alpha(t)\|^2\, \mathrm{d}t\ . \tag{1}$$

Here, $\dot\alpha$ denotes the tangent of α and $\mathcal{D}$ the domain of its argument. Furthermore, $\chi_\varphi\colon \mathcal{S} \to \mathcal{S}_\varphi$ is a smooth diffeomorphism and, consequently, for every curve $\tilde\alpha \in \Gamma_{\gamma_\varphi}(\mathcal{S}_\varphi)$ there is a curve $\alpha \in \Gamma_\gamma(\mathcal{S})$ with $\tilde\alpha = \chi_\varphi \circ \alpha$. Using this, the optimization problem can be reformulated to describe our whole algorithm:

$$\begin{gathered}\underset{\alpha \in \Gamma_{\gamma}(\mathcal{S})}{\arg\min}\, E(\chi_\varphi \circ \alpha) \quad \text{with} \\ E(\chi_\varphi \circ \alpha) = \frac{1}{2}\int_{\mathcal{D}(\alpha)} \|\dot\alpha(t)\|^2 + |\langle \nabla\varphi \circ \alpha(t) \mid \dot\alpha(t)\rangle|^2\, \mathrm{d}t\ .\end{gathered} \tag{2}$$

The magnitude of $\nabla\varphi$ is expected to take higher values for points that exhibit larger distances to the initial curve. Although traversing parts of φ's level sets does not result in an energy increase regardless of the distance to the initial curve, changing level sets generates a cost at least for *open* curves. This is because the curve must travel along the direction of $\nabla\varphi$ to reach a different level set.

The situation is different in the case of *closed* curves, where there are no fixed endpoints and the same cannot be assumed. It is possible to construct pathological cases with geodesics as level sets in which a local solution to the optimization problem leads to geodesics that are arbitrarily far away from the original. These cases are, however, unlikely and structurally unstable as a small perturbation of the potential resolves the situation. Also, the algorithm employed for geodesic tracing is a local method. Consequently, it is more likely to converge to a closer solution even in the presence of pathological cases.

Based on this discussion, the energy increases for both, *open and closed* curves, that are further away from the initial curve. However, the first term in the integrand still leads to a local shortening and will, therefore, also reduce geodesic curvatures, although not in a uniform sense, which fits our understanding of smoothing. As the output λ of our algorithm also represents a local minimizer to the above optimization problem with the initial guess γ, this formulation makes apparent that it will need to locally shorten and reduce traveling at the same time. Hence, a result from the algorithm must achieve a balance between smoothness and closeness to the initial curve.

Moreover, this formulation validates the earlier explanation of the penalty potential as a distance-based constraint and the tracing of locally shortest geodesics as the smoothing strategy. This indicates that employing the fourth dimension is not an obligatory prerequisite, as we can resort to the optimization problem as an alternative formulation. Nevertheless, lifting into 4D Euclidean space gives an alternative and intuitive explanation. And most importantly, lifting transforms the numerical optimization problem to a significantly simpler problem: the computation of geodesic paths, for which efficient, robust, and practical implementations exist. In addition, lifting provides the potential for algorithmic extensions.

3.3. Construction of Penalty Potentials

The construction of the penalty potential poses the primary challenge in implementing our algorithm, and there are numerous reasonable choices. Our goal is to present a one-parameter family of potentials, allowing for the adjustment of the trade-off between smoothness and similarity to the initial curve. This ensures some flexibility of the algorithm while maintaining an intuitive user interface. We note this generic potential function works well for a wide range of applications but is not tailored to specific scenarios. However, this can be done easily by adapting our family of potential functions or designing new ones. In this sense, the following construction can serve as a template and guide for creating individual penalty potentials.

The potential φ penalizes points further from the initial curve by mapping them to higher values. According to the optimization formulation in Section 3.2, this must be accomplished in a manner that ensures $\|\nabla\varphi\|$ also increases for higher distances. For that, let $d_\gamma\colon \mathcal{S} \to [0,\infty)$ be the geodesic distance field that assigns every point on the surface its minimal geodesic distance to the initial curve. Unfortunately, d_γ is in general not smooth [CWW13], which poses a challenge to our approach of describing everything in terms of operations on smooth manifolds. Luckily, a smooth distance field δ_γ can be obtained as a by-product by employing the heat method with larger time steps [CWW13]. However, the use of the heat method is not mandatory. In cases where it is not used, we recommend using a convolution with a normalized bump function $h \in C_c^\infty(\mathcal{S})$, similar to Laplacian smoothing as proposed by LIVESU [Liv18]:

$$\delta_\gamma(\mathbf{x}) := \int_{\mathcal{S}} h(\mathbf{y})\, d_\gamma(\mathbf{x}-\mathbf{y})\, \mathrm{d}\sigma(\mathbf{y}) \ , \tag{3}$$

which is simple and efficient. Alternatively, more advanced methods can be used. Using general statements from calculus, the result of a convolution inherits various properties from its arguments. δ_γ must be smooth as h is a smooth function. On the other hand, the small and compact support ensures that δ_γ only slightly changes in comparison to d_γ. In practice, the non-smoothness of d_γ mostly affects the computation of derivatives and, consequently, is negligible in this particular family of potentials. For potentials that rely on the gradient of the distance field, smoothness is an essential requirement and should be accomplished by one of the mentioned methods.

The smooth distance field δ_γ itself is not sufficient as a penalty potential. The intrinsic nature of a geodesic distance field dictates that $\|\nabla\delta_\gamma\| \approx 1$ does not increase with higher distance. We fix this issue by introducing a family of smooth modifier functions $m_\tau\colon [0,\infty) \to [0,\infty)$ that is parameterized by a tolerance value $\tau \in [0,\infty)$ and simply define φ to be the composition of a chosen modifier function and the smoothed distance.

$$\varphi := m_\tau \circ \delta_\gamma \tag{4}$$

Our choice for m_τ composes two functions $f,g\colon [0,\infty) \to [0,\infty)$:

$$m_\tau(x) := g(a\tau x) \qquad g(x) := L \cdot f\left(\frac{x}{L}\right) \qquad f(x) := x^2 \tag{5}$$

$$a := 10 \qquad L := \|\delta_\gamma\|_\infty \tag{6}$$

As these choices may seem arbitrary and ad-hoc, we give the following rationale: For the definition of m_τ, there exist various options that accomplish the objective. To narrow options, we consider additional properties and trade-offs such as simplicity, efficiency, numerical robustness, and mesh independence.

In this regard, f is one of the simplest functions that can ensure a decent increase of $\|\nabla\varphi\|$ for larger distances from the initial curve. Other strictly monotone functions, e.g., such as x^n for $n > 2$ or e^x, are also valid choices but may more quickly lead to numerical failure for finding geodesics in the lifted surface when distances get too large. Also, one multiplication is an inexpensive operation.

For τ, the adjustment between smoothness and proximity of the resulting curve shall lead to similar results across various surface mesh scales. To achieve this, g incorporates a characteristic length L that may depend on the surface mesh and the initially given curve. The idea is to scale the input distance and the resulting potential value equally. In our test cases, we set $L := \|\delta_\gamma\|_\infty$, i.e., the maximum over all possible values of δ_γ. This choice yields uniform outcomes across our range of different surface meshes. An alternative would be to consider $L := \mathrm{L}(\gamma)$, i.e., the length of the initial curve. In this case, it can be observed that longer curves will be affected by stronger smoothing, whereas shorter curves will stick more closely to their original shape. This can result in a more desirable behavior for some applications.

The final component of our construction involves the scaling of the argument of the potential by τ to allow for a more restrictive or relaxed potential. This again is a particularly simple choice to achieve this behavior that also allows for fast computation without introducing large numerical errors. When $\tau = 0$, the resulting potential is zero, and the smoothed curve describes a geodesic. As τ is increased, the curve is constrained to traverse closer to the original. Additionally, we introduce a normalization constant a to provide a superior default trade-off when $\tau = 1$. This particular choice is arbitrary and would most likely need to be changed for a different characteristic length. We remark that this choice is not crucial for the construction.

4. Implementation

We represent surfaces as triangle meshes and surface curves as piecewise linear curves with vertices on surface edges or faces. The essential steps of our algorithm are the construction of a smoothed distance field and finding a locally shortest geodesic in the lifted surface. Both problems have been studied extensively for discrete geometry representation, and there exist robust and efficient algorithms with publicly available implementations. We build upon prior work: For our implementation, we chose *Geodesics in Heat (The Heat Method)* [CWW13] and *FlipOut* [SC20]. We emphasize that this choice is not unique, and alternative methods can be employed as well (refer to surveys mentioned in Section 2). The only requirement for any method is being able to measure (edge) lengths on a given oriented topological 2-manifold.

For the construction of a smoothed distance field δ_γ, we use the implementation of the heat method [CWW13] in *Libigl* [JP*18]. The use of *Libigl* does not constrain the choice of data structures for the surface mesh or the surface mesh curve. In brief, the heat method proceeds in two steps. First, an initial heat flow is integrated for a given time step. The source of this flow is given by the curve from which the distance is measured. Second, the method finds a scalar field whose gradient approximates the normalized gradient of the heat flow in a least-squares sense. This leads to solving a sparse linear system. Since time steps control the amount of diffusion in the first step, this heat method yields smoothed distance fields as a side effect when using large time steps. As mentioned in Section 3.3, for potentials where the distance field's smoothness is essential, this eliminates the need for post-processing, e.g., convolution-based smoothing.

To identify locally shortest geodesics within a given isotopy class, we use the *FlipOut* algorithm [SC20] in the implementation provided by *Geometry Central* [SC*19]. The implementation relies on intrinsic triangulations and the *signpost* data structure [SSC19]. For this reason, we use the surface mesh data structure provided by *Geometry Central* to represent triangle meshes. *FlipOut* expects the initial surface mesh curve to be a simple vertex curve in the intrinsic triangulation. This is not a restriction as we can freely add other points on edges or even inside faces as new intrinsic vertices. In essence, *FlipOut* leverages intrinsic edge flips to iteratively shorten the given curve. The resulting curves are locally shortest geodesics in the discrete setting and approximate smooth geodesics. The algorithm is efficient, it offers a robust handling of open and closed curves as well as several edge cases, and termination is guaranteed as there is no convergence in a continuous sense involved. As a further benefit, the algorithm can be configured to prevent the shrinking of closed loops to single points by specifying a minimal scale of length reduction.

In our implementation, the initial surface curve γ is determined by user interaction similar to LAWONN et al. [LGRP14]: The user adds points on the surface mesh by literally painting a line. Each pair of neighboring points is then connected by the shortest path on the surface mesh, resulting in an initial jagged curve. To improve overall robustness, we filter the initial curve by *capping corners*: If two consecutive segments run through one triangle they are replaced by the single line segment spanned by the opposite edge. We remark that the particular origin of the initial curve is not important to our method. It may be generated differently, e.g., from (semi-)automatic methods like feature extraction or feature-based segmentation [LGRP14; Liv18].

The implementation of the heat method in *Libigl* computes the average edge-length of the surface mesh, to determine the time step, and sets up the mass and stiffness matrices [CWW13; JP*18]. This is a preprocessing step that is independent of the input curve. Optionally, we enlarge the time step by a specific scale to be able to obtain a smoothed distance field. After that, each vertex of the initial curve γ is provided as a source to the heat method computation to determine the smoothed distance δ_γ as a piecewise linear function by assigning every vertex a distance value. Subsequently, the characteristic length can be calculated and the modifier function m_τ can be applied directly to the values of all vertices (see Section 3.3). Our implementation computes the distance field on the whole mesh. For large meshes, one could alternatively determine a region of interest, such that all computations are restricted to this subset of the surface, similar to the distance envelope used by LAWONN et al. [LGRP14].

There is no need for an explicit representation of the lifted surface mesh $\mathcal{S}_\varphi$. Instead, we provide a function to compute edge lengths, which uses lifted vertex positions, thus changing the surface metric. For two adjacent vertices $\mathbf{p}, \mathbf{q} \in \mathcal{S}$, their lifted edge distance $g_\varphi(\mathbf{p}, \mathbf{q})$ is given as:

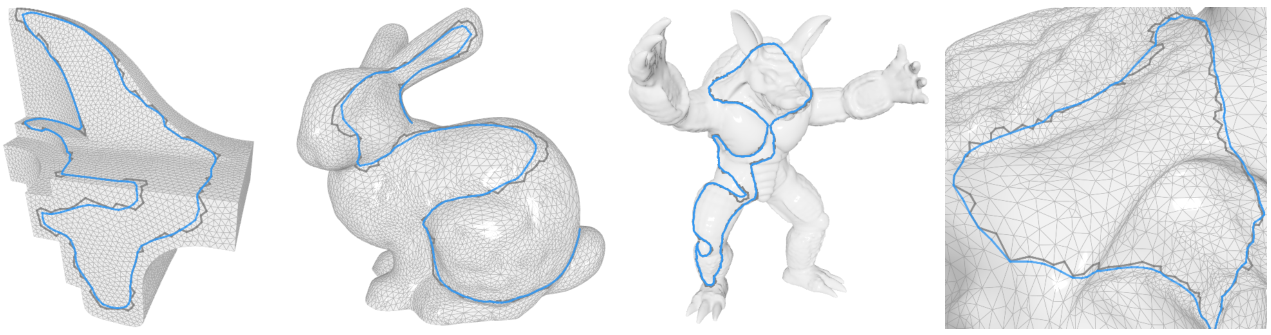
$$g_\varphi(\mathbf{p}, \mathbf{q}) := \sqrt{\|\mathbf{p} - \mathbf{q}\|^2 + |\varphi(\mathbf{p}) - \varphi(\mathbf{q})|^2}\,. \tag{7}$$

Applying the *FlipOut* algorithm to this altered edge distance, it automatically determines the topology of the locally shortest geodesic. The last step of the algorithm is the projection operation and consists of gathering all points on edges and vertices and to compute their actual position by using the original 3D geometry.

In the previous section, we operated on smooth manifolds and, as with the heat method, we must address the numerical errors arising from a discrete surface mesh. We neglect approximation errors originating from the heat method which can be resolved by adaptive refinement. On the other hand, φ is only approximated by a piecewise linear function. The smoothing process, however, relies on a smooth change of curvature around the initial curve γ in the lifted surface $\mathcal{S}_\varphi$ to ensure that the projected geodesic λ is an actual smoother version. As $\nabla\varphi$ will be constant across each face, a (too) coarse triangulation must lead to discretization artifacts. Especially in areas of larger amounts of negative curvatures in the lifted surface $\mathcal{S}_\varphi$, $\nabla\varphi$ may fluctuate. This would result in a loss of smoothness for the projected geodesic λ that manifests in the form of small noise (see Figure 9).

Table 1: *Performance results for various surface mesh models. n_0 is the number of vertices of the initial curve. n is the number of vertices of the resulting curve. t_φ and t_λ denote the time need to determine the penalty potential and finding the geodesic, respectively. $t = t_\varphi + t_\lambda$ is the overall running time. Figures 1, 3, 4, and 7 illustrate the models and curves corresponding to this table.*

Model	Faces	τ	n_0	n	t [ms]	t_φ [ms]	t_λ [ms]
Fandisk	∼ 14,000	2.0	257	440	12	2	10
Bunny	∼ 15,000	1.0	196	325	13	4	9
Armadillo	∼ 86,000	2.0	555	949	52	7	44
Dragon	∼ 124,000	1.5	548	951	80	27	53
Sappho's Head	∼ 282,000	2.5	698	1225	189	70	118
Grumpy Pumpkin	∼ 395,000	1.8	1258	2202	362	163	199
Brain	∼ 554,000	1.5	1655	3061	453	171	282
Fox Skull	∼ 700,000	1.5	789	1512	589	227	362
Fairing	∼ 994,000	1.5	2007	3996	1070	450	620



(a) *Fandisk* **(b)** *Bunny* **(c)** *Armadillo* **(d)** *Armadillo Zoom Left Shoulder*

Figure 3: *Models with initial (black) and smoothed (blue) curves. These images correspond to the data shown in table 1.*

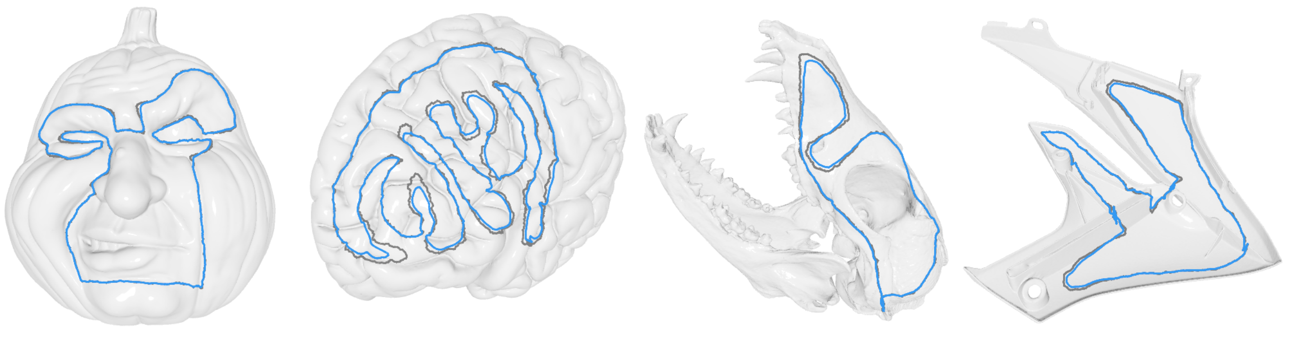
These artifacts from inappropriate tessellation can be fixed by an optional post-processing following the constrained Laplacian smoothing proposed in LAWONN et al. [LGRP14]. After the projection of the lifted geodesic, the curve is already close to the desired solution. Thus, the post-process requires only a few iterations of relaxation and there are no issues with convergence. We relax only on the mesh edges supporting the curve and don't apply splitting and merging of curve vertices.

5. Results and Discussion

We apply our method for a number of triangle meshes and interactively painted surface curves. Table 1 provides an overview of the data sets, timings, and parameters τ. All times reported in figures and tables were measured using a single core of the Intel Core i9-13900K CPU on a machine with 32 GiB RAM. The initial discrete surface curves are created by the painting process described in Section 4. To simulate noise, we restrict the initial curves such that they are spanned by vertices of the mesh, i.e., curves intersect edges only at their end points. We run tests on different surface meshes of varying sizes ranging from the order of 10 000 to near a million faces. The benchmark surfaces show varying degree of smoothness and local detail. The results are shown in Figures 1, 3, 4, and 7. In all experiments, we used different but similar open and closed curves and observed no significant differences in performance or quality. The figures only ever depict one exemplary curve. For each surface mesh, the initial time steps of the heat method, given by the average edge length, were scaled by a factor of 10 to obtain smoother distance fields.

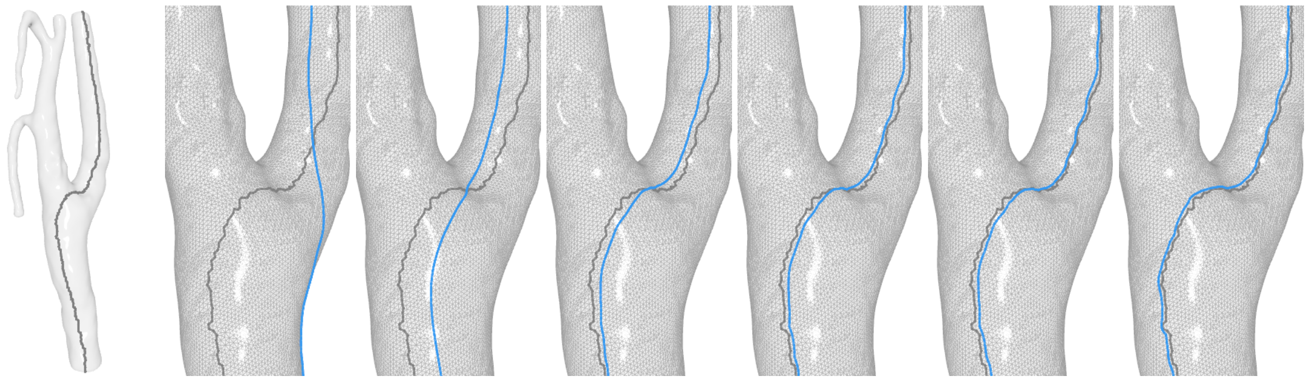
In summary, we achieve interactive frame rates for most examples and are near interactive for the meshes with more than 500 000 faces. For all examples, the smoothing increases the number of curve vertices by less than a factor of two. The figures show that the initial curves are effectively smoothed. The reduction of noise and artifacts (also in the absence of curvature) can be confirmed visually. The deviation of the smoothed curve from the initial curve is effectively controlled by the parameter τ.

Performance Interestingly, for the small to medium size meshes, the time for construction of penalty potentials is lower than the time for tracing geodesics. Only for large meshes with millions of triangles, the heat method becomes the bottleneck of our approach. This could be remedied by restricting the curve and hence the penalty potential to a submesh that covers the original curve. The heat method uses a sparse Cholesky factorization that in theory offers a sub-quadratic complexity for Poisson-type problems. In practice, this is often better and roughly linear in the number of vertices. Alternatively, iterative solvers such as the conjugate gradient method could be applied. [CWW13]. On the contrary, *FlipOut*'s



(a) *Grumpy Pumpkin* **(b)** *Brain* **(c)** *Fox Skull* **(d)** *Fairing*

Figure 4: *High-resolution models with initial (black) and smoothed (blue) curves. These images correspond to the data shown in Table 1.*



(a) *Initial* **(b)** $\tau = 0.0$ **(c)** $\tau = 0.5$ **(d)** $\tau = 1.0$ **(e)** $\tau = 1.5$ **(f)** $\tau = 2.0$ **(g)** $\tau = 2.5$

Figure 5: *Influence of the parameter* τ*. The images show the Carotid Artery with initial (black) smoothed (blue) curves for varying parameters* τ*. Each application of the algorithm took approximately* $\sim 42-47\,\text{ms}$*. The construction of* φ *took* $\sim 20\,\text{ms}$ *and the tracing of* λ_φ *took* $\sim 22-27\,\text{ms}$ *with smaller* τ *resulting in larger values.*

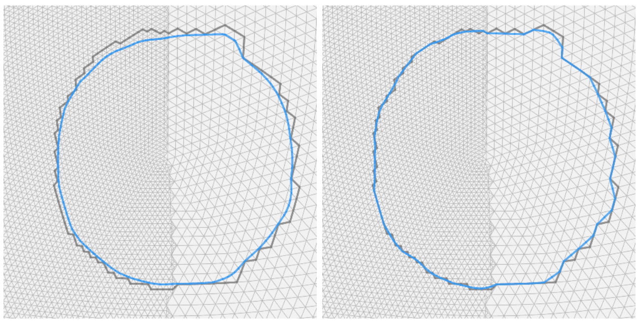
complexity is sub-linear [SC20] which explains the break-even point of computation times.

Effect of Tolerance Parameter The parameter study in Figure 5 visualizes the effect of varying τ for the *Carotid Artery* model: For small τ, the output curve can move more freely on the surface and is smoother, while large values constrain the output close to the initial curve. As described in Section 3.3, for $\tau = 0$, the algorithm yields a locally shortest geodesic in the same isotopy class. According to our conceptualization of smoothing, this is one of the smoothest curves possible. However, it is also evident that such a geodesic can be located at a considerable distance from the initial curve. For values of τ that are exceedingly large, it is not possible to adhere to the initial curve's shape with absolute precision. This is because the penalty potential φ would have to converge to infinity for points that are not part of the curve, while remaining zero at curve vertices. Mathematically, this case may not even exist, as it would contravene the property of smoothness. In practice, numerical errors prevent this possibility at all.

The variation of τ only has a slight effect on the overall run times. Smaller values result in an increase of time to trace the geodesic. This is expected, as the computation of the potential is independent of τ, and only the tracing of geodesic curves is affected. As *FlipOut* is working with intrinsic edge flips, a smaller tolerance value will give the curve more freedom and, consequently, result in more edge flips to reach the final result.

Robustness and Comparison Figures 6 and 7 illustrate our curve smoothing approach for surface meshes with varying spatial discretization for planar and more complex geometry. From right to left, the surface mesh resolution of the *Dragon* scene from Figure 1 has been modified to get continuously higher. The *Circle* incorporates a sudden change in its face resolution. In both cases, comparable degrees of smoothing are achieved under identical parameter settings. For coarser triangulations, the constraint associated with the potential φ is somewhat less stringent, as φ is only a piecewise linear approximation across vertices.

Also, we compared our implementation to the hyper surface smoothing of LIVESU [Liv18] (see Figures 8 and 9). For a fair comparison, we used its original implementation in our code. In all cases, our algorithm was several times faster than the hyper surface smoothing. Furthermore, as the hyper surface smoothing is mainly applied in the context of mesh segmentation, its input is restricted to closed curves that partition the surface into two distinct subsets. The



(a) $\tau = 1.5$ **(b)** $\tau = 5.0$

Figure 6: *Robustness for Circle scene with a sudden resolution change, initial curve (black), and smoothed curve (blue).*

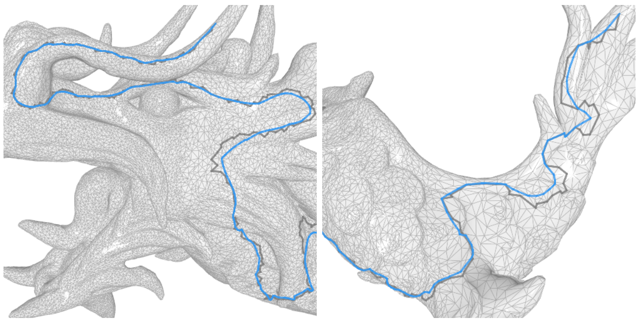


Figure 7: *Robustness for the Dragon with continuous resolution change, initial curve (black), and smoothed curve (blue).*

algorithm's output is not directly given as another surface curve but instead implicitly returned by a scalar field. The actual smoothed curve can only be evaluated by tracing the 0-level set of this field. The level set is mathematically not required to fulfill the conditions of a single closed surface curve. Indeed, in Figure 8, we could easily construct a case for the *Armadillo* scene for which this approach is not robust and not returning meaningful results. Also, for the *Cow* scene, a strong bias of the solution due to varying face sizes could be observed. For each valid result produced by hyper surface smoothing, we observed that our approach offers a comparable smoothing quality. In conclusion, our approach produces robust results for all tested cases more efficiently (i.e. within a smaller amount of time) with comparable quality.

Post-Processing For high-resolution meshes, differences arising from using the optional Laplace relaxation post-processing are barely visible, i.e., there is no need for post-processing. On the other hand, curves on coarsely triangulated meshes may significantly benefit. On the left side in Figure 9, noise artifacts can be observed for coarse triangulations in our smoothed curve without using a post-processing filter. In this case, the quality of hyper surface smoothing by LIVESU [Liv18] offers a higher quality. The right side shows a much smoother (blue) curve with post-processing turned on. Please note that the underlying triangle strip of our smoothed curve does not change and now again offers a comparable quality. With respect to performance measurements, the post-processing

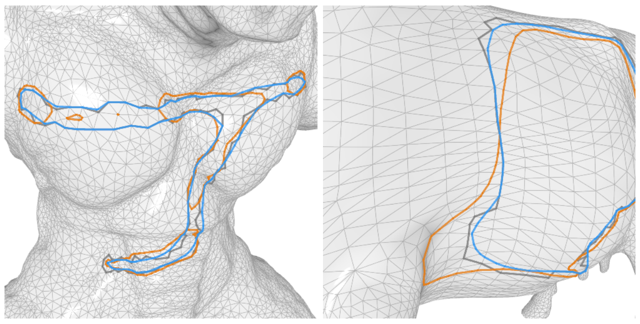


Figure 8: *Robustness comparison of our curve smoothing approach (blue) with* LIVESU *[Liv18] (orange) based on the same initially given curve (black).*

(less than 1 % of the overall running time) can be neglected. In all our experiments, there were no significant differences after only 5 iterations of relaxation.

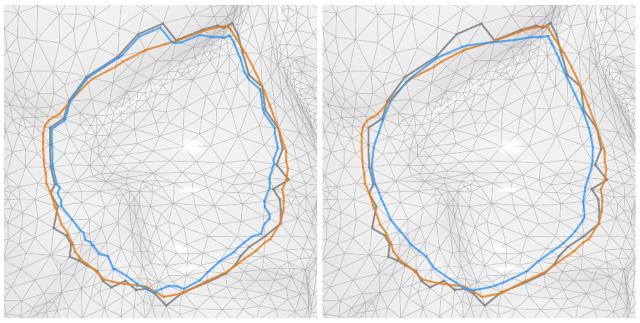


Figure 9: *Effect of post-processing for coarse triangulations with initial curve (black), smoothed curve (blue), and* LIVESU *[Liv18] (orange). Post-processing is switched off (left) and on (right).*

Applications We see potential use of our method in medical applications, which are also emphasized in LAWONN et al. [LGRP14]. One particular use case is the detection and segmentation of aneurysms: LAWONN et al. [LMW*19] presented a method for identifying triangles that either belong to an aneurysm or to the remaining vessel parts. This resulted in a partition of the surface that was represented by a scalar field. The authors then applied a smoothing scheme to this scalar field in order to achieve a smoother curve that separates the two subsets. The method identifies the smoothed curve as a level set of the smoothed potential. Our algorithm, seen in Figure 10, allows for the robust and direct smoothing of the underlying curve itself, in contrast to the scalar field. A second application is given by EULZER et al. [ERM*21], who automatically cut and flatten vascular geometry to obtain an intuitive mapping between the 3D and 2D domains. Their automatic cuts are only represented as edge curves. To improve intuition behind the obtained mapping, our algorithm could be employed to smooth these cuts. Figure 5 shows an exemplary *Carotid Artery* surface mesh [EvDH*23; ERP*24]. Additionally, the right side of Figure 10 also illustrates the application of our algorithm to a more complicated blood vessel.

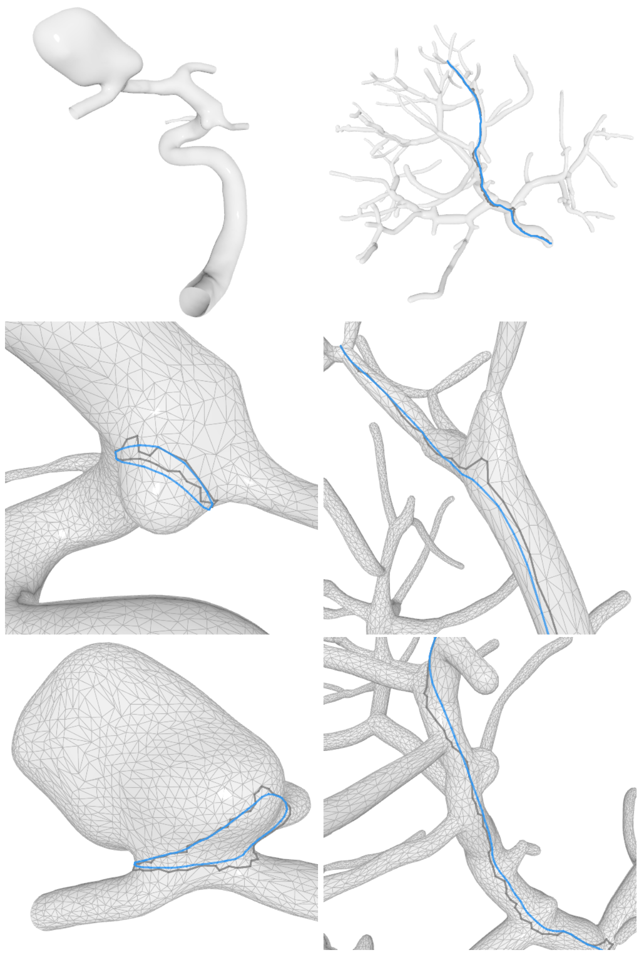


Figure 10: *Application to the segmentation of aneurysms (left) and blood vessels (right) with initial (black) and smoothed (blue) curves, rendered without hidden line removal for better visibility.*

Further Limitations Generally, our method suffers from inappropriate and/or coarse tessellations. The heat method may generate (locally) unfeasible distance fields, and the surface mesh itself may be too coarse to "support" a smooth curve that consists of vertices on triangle edges. As shown on the left side in Figure 9, this manifests in noise artifacts in the resulting curve. The post-processing can leverage this but only to a certain degree. It can also be observed that by definition our algorithm can only shorten curves which sometimes leads to a sub-optimal result whose shape is not following its original close enough. This effect also originates from another problem: while it is evident that, with the use of an appropriate penalty potential, a smoothed curve generated by our algorithm must stay in close proximity to the initial curve, the opposite is less clear. In fact, we cannot prove that for all points of the initial curve, a part of the smoothed curve must pass nearby. Hence, our approach is limited as it only controls a "one-sided Hausdorff distance". Indeed, our findings indicate that certain corners of the initial curves are occasionally omitted. Our current implementation cannot handle surface curves that self-intersect. This limitation arises from the chosen algorithm for tracing geodesics, not the overall method. We note, however, that input curves with self-intersections and curves with parts of them getting close to each – the extreme is a space filling curve – may result in penalty potentials that give too much freedom to the smoothed curve. Such pathological cases require a tessellation that is dense enough to support the curve and regions between parts of the curve as well as a careful local choice of τ.

6. Conclusions and Future Work

We presented a new concise algorithm to robustly smooth discrete surface curves on triangular surface meshes that is able to adjust the trade-off between smoothness and similarity to the initial curve. Our method is based on constructing a potential on the surface that penalizes larger distances from the initial curve with larger values. The potential is used to lift and embed the surface in 4D Euclidean space to determine a geodesic that represents the smoothed curve when projected onto the original surface in 3D.

We show that our approach can be interpreted as a reformulation of a continuous energy minimization with non-trivial constraints as the curve is part of the surface. The benefit of this reformulation is an efficient and robust non-iterative algorithm that builds upon well established building blocks for computing geodesic distance fields and geodesics. The first requires only the solution of a sparse linear system using the heat method. And the second typically uses a discrete algorithm with guaranteed termination like *FlipOut*. The implementation of our method is straightforward as we can take advantage of existing libraries. We propose a one-parameter family of penalty potentials. Our experiments show that this works effectively and efficiently for a wide range of surface meshes. The effect of parameter variation is intuitive, and the result can be computed at interactive frame rates, which enables variation with instant visual feedback.

Future work includes further research into the construction of alternative penalty potentials, possibly even complex-valued, to achieve more advanced morphing of curves. We also want to allow for more interaction with the curve by adding constraints and local intuitive adjustments to the penalty potential for certain meshes. In regard to the main algorithmic limitations, our algorithm can only shorten curves and does not necessarily reduce the Hausdorff distance between input and result. Regarding the main limitations of our implementation, discretization errors from coarse triangulations cause noise artifacts that need to be handled by further post-processing routines. Furthermore, our implementation can only handle non-crossing surface curves. This is due to the *FlipOut* algorithm used to determine the lifted (locally shortest) geodesic, which cannot handle self-intersecting curves [SC20]. To address this issue, an alternative algorithm could be used to generate geodesics. To enhance our implementation's performance and introduce upper bounds for the smoothed curve's deviation, we suggest to limit the surface to a smaller sub-mesh, so-called distance envelope [LGRP14]. Nevertheless, we believe that this work provides a new and elegant curve smoothing tool for several application domains, such as illustrative visualization and surgical planning.

References

[AA81] ALEXANDER, RALPH and ALEXANDER, S. "Geodesics in Riemannian Manifolds-with-Boundary". *Indiana University Mathematics Journal* 30.4 (1981), 481–488. JSTOR: 24893014 4.

[ABB87] ALEXANDER, STEPHANIE B., BERG, I. DAVID, and BISHOP, RICHARD L. "The Riemannian Obstacle Problem". *Illinois Journal of Mathematics* 31.1 (1987), 167–184. DOI: 10.1215/ijm/1255989406 4.

[AR20] ALIRR, OMAR IBRAHIM and RAHNI, ASHRANI AIZZUDDIN ABD. "Survey on Liver Tumour Resection Planning System: Steps, Techniques, and Parameters". *Journal of Digital Imaging* 33.2 (2020), 304–323. DOI: 10.1007/s10278-019-00262-8 1.

[BLVD11] BENHABILES, HALIM, LAVOUÉ, GUILLAUME, VANDEBORRE, JEAN-PHILIPPE, and DAOUDI, MOHAMED. "Learning Boundary Edges for 3D-Mesh Segmentation". *Computer Graphics Forum* 30.8 (2011), 2170–2182. DOI: 10.1111/j.1467-8659.2011.01967.x 1.

[BMSW11] BOSE, PROSENJIT, MAHESHWARI, ANIL, SHU, CHANG, and WUHRER, STEFANIE. "A Survey of Geodesic Paths on 3D Surfaces". *Computational Geometry* 44.9 (2011), 486–498. DOI: 10.1016/j.comgeo.2011.05.006 2.

[BWK05] BISCHOFF, STEPHAN, WEYAND, TOBIAS, and KOBBELT, LEIF. "Snakes on Triangle Meshes". *Bildverarbeitung Für Die Medizin 2005*. Ed. by MEINZER, HANS-PETER, HANDELS, HEINZ, HORSCH, ALEXANDER, and TOLXDORFF, THOMAS. Berlin/Heidelberg: Springer-Verlag, 2005, 208–212. ISBN: 978-3-540-25052-4. DOI: 10.1007/3-540-26431-0_43. URL: http://link.springer.com/10.1007/3-540-26431-0_43 3.

[CLPQ20] CRANE, KEENAN, LIVESU, MARCO, PUPPO, ENRICO, and QIN, YIPENG. *A Survey of Algorithms for Geodesic Paths and Distances*. 2020. arXiv: 2007.10430 [cs]. URL: http://arxiv.org/abs/2007.10430. preprint 2, 4.

[CWW13] CRANE, KEENAN, WEISCHEDEL, CLARISSE, and WARDETZKY, MAX. "Geodesics in Heat". *ACM Transactions on Graphics* 32.5 (2013), 1–11. DOI: 10.1145/2516971.2516977. arXiv: 1204.6216 [cs] 3, 5–7.

[DAZ*20] DU, XINGYI, AIGERMAN, NOAM, ZHOU, QINGNAN, et al. "Lifting Simplices to Find Injectivity". *ACM Transactions on Graphics* 39.4 (2020), 120:120:1–120:120:17. DOI: 10.1145/3386569.3392484 3.

[DKZ*21] DU, XINGYI, KAUFMAN, DANNY M., ZHOU, QINGNAN, et al. "Optimizing Global Injectivity for Constrained Parameterization". *ACM Transactions on Graphics* 40.6 (2021), 260:1–260:18. DOI: 10.1145/3478513.3480556 3.

[DKZ*22] DU, XINGYI, KAUFMAN, DANNY M., ZHOU, QINGNAN, et al. "Isometric Energies for Recovering Injectivity in Constrained Mapping". *SIGGRAPH Asia 2022 Conference Papers*. SA '22. New York, NY, USA: Association for Computing Machinery, 2022, 1–9. ISBN: 978-1-4503-9470-3. DOI: 10.1145/3550469.3555419. URL: https://dl.acm.org/doi/10.1145/3550469.3555419 3.

[ERM*21] EULZER, PEPE, RICHTER, KEVIN, MEUSCHKE, MONIQUE, et al. *Automatic Cutting and Flattening of Carotid Artery Geometries*. The Eurographics Association, 2021. ISBN: 978-3-03868-140-3. URL: https://doi.org/10.2312/vcbm.20211347 9.

[ERP*24] EULZER, PEPE, RICHTER, KEVIN, PROBST, TRISTAN, et al. *A Dataset of Reconstructed Carotid Bifurcation Lumen and Plaque Models with Centerline Tree and Simulated Hemodynamics*. Version 2.0.0. Zenodo, 2024. DOI: 10.5281/zenodo.10695923. URL: https://zenodo.org/records/10695923 9.

[EvDH*23] EULZER, P., von DEYLEN, F., HSU, W.-C., et al. "A Fully Integrated Pipeline for Visual Carotid Morphology Analysis". *Computer Graphics Forum* 42.3 (2023), 25–37. DOI: 10.1111/cgf.14808 9.

[HP04] HOFER, MICHAEL and POTTMANN, HELMUT. "Energy-Minimizing Splines in Manifolds". *ACM Transactions on Graphics* 23.3 (2004), 284–293. DOI: 10.1145/1015706.1015716 2.

[HPY21] HA, YUJIN, PARK, JUNG-HO, and YOON, SEUNG-HYUN. "Geodesic Hermite Spline Curve on Triangular Meshes". *Symmetry* 13.10 (10 2021), 1936. DOI: 10.3390/sym13101936 2.

[JK04] JUNG, MOONRYUL and KIM, HAENGKANG. "Snaking across 3D Meshes". *12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings.* 12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings. Seoul, Korea: IEEE, 2004, 87–93. ISBN: 978-0-7695-2234-0. DOI: 10.1109/PCCGA.2004.1348338. URL: http://ieeexplore.ieee.org/document/1348338/ 3.

[JLCW06] JI, ZHONGPING, LIU, LIGANG, CHEN, ZHONGGUI, and WANG, GUOJIN. "Easy Mesh Cutting". *Computer Graphics Forum* 25.3 (2006), 283–291. DOI: 10.1111/j.1467-8659.2006.00947.x 1.

[JP*18] JACOBSON, ALEC, PANOZZO, DANIELE, et al. *Libigl*. 2018. URL: https://libigl.github.io/ 6.

[JSW*19] JIN, YAO, SONG, DAN, WANG, TONGTONG, et al. "A Shell Space Constrained Approach for Curve Design on Surface Meshes". *Computer-Aided Design* 113 (2019), 24–34. DOI: 10.1016/j.cad.2019.03.001 2.

[KT09] KAPLANSKY, LOTAN and TAL, AYELLET. "Mesh Segmentation Refinement". *Computer Graphics Forum* 28.7 (2009), 1995–2003. DOI: 10.1111/j.1467-8659.2009.01578.x 1.

[LGRP14] LAWONN, KAI, GASTEIGER, ROCCO, RÖSSL, CHRISTIAN, and PREIM, BERNHARD. "Adaptive and Robust Curve Smoothing on Surface Meshes". *Computers & Graphics* 40 (2014), 22–35. DOI: 10.1016/j.cag.2014.01.004 1–3, 6, 7, 9, 10.

[Liv18] LIVESU, MARCO. "A Heat Flow Based Relaxation Scheme for *n* Dimensional Discrete Hyper Surfaces". *Computers & Graphics* 71 (2018), 124–131. DOI: 10.1016/j.cag.2018.01.004 1–3, 5, 6, 8, 9.

[LMW*19] LAWONN, KAI, MEUSCHKE, MONIQUE, WICKENHÖFER, RALPH, et al. "A Geometric Optimization Approach for the Detection and Segmentation of Multiple Aneurysms". *Computer Graphics Forum* 38.3 (2019), 413–425. DOI: 10.1111/cgf.13699 9.

[LVPI18] LAWONN, KAI, VIOLA, IVAN, PREIM, BERNHARD, and ISENBERG, TOBIAS. "A Survey of Surface-Based Illustrative Rendering for Visualization: Surface-Based Illustrative Rendering". *Computer Graphics Forum* 37.6 (2018), 205–234. DOI: 10.1111/cgf.13322 1.

[LZH*07] LAI, YU-KUN, ZHOU, QIAN-YI, HU, SHI-MIN, et al. "Robust Feature Classification and Editing". *IEEE Transactions on Visualization and Computer Graphics* 13.1 (2007), 34–45. DOI: 10.1109/TVCG.2007.19 3.

[MCV07] MORERA, DIMAS MARTINEZ, CARVALHO, PAULO CEZAR, and VELHO, LUIZ. "Geodesic Bezier Curves: A Tool for Modeling on Triangulations". *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*. XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007). Minas Gerais, Brazil: IEEE, 2007, 71–78. ISBN: 978-0-7695-2996-7. DOI: 10.1109/SIBGRAPI.2007.38. URL: http://ieeexplore.ieee.org/document/4368170/ 2.

[MNPP23] MANCINELLI, CLAUDIO, NAZZARO, GIACOMO, PELLACINI, FABIO, and PUPPO, ENRICO. "B/Surf: Interactive Bézier Splines on Surface Meshes". *IEEE Transactions on Visualization and Computer Graphics* 29.7 (2023), 3419–3435. DOI: 10.1109/TVCG.2022.3171179 2–4.

[MVC05] MARTÍNEZ, DIMAS, VELHO, LUIZ, and CARVALHO, PAULO C. "Computing Geodesics on Triangular Meshes". *Computers & Graphics* 29.5 (2005), 667–675. DOI: 10.1016/j.cag.2005.08.003 4.

[MVC08] MORERA, DIMAS MARTINEZ, VELHO, LUIZ, and CARVALHO, PAULO CEZAR. "Subdivision Curves on Surfaces and Applications". *Progress in Pattern Recognition, Image Analysis and Applications*. Ed. by RUIZ-SHULCLOPER, JOSÉ and KROPATSCH, WALTER G. Vol. 5197. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, 405–412. ISBN:

978-3-540-85919-2 978-3-540-85920-8. DOI: 10.1007/978-3-540-85920-8_50. URL: http://link.springer.com/10.1007/978-3-540-85920-8_50 2.

[Pat16] PATANÉ, GIUSEPPE. "STAR - Laplacian Spectral Kernels and Distances for Geometry Processing and Shape Analysis". *Computer Graphics Forum* 35.2 (2016), 599–624. DOI: 10.1111/cgf.12866 2.

[PH05] POTTMANN, HELMUT and HOFER, MICHAEL. "A Variational Approach to Spline Curves on Surfaces". *Computer Aided Geometric Design* 22.7 (2005), 693–709. DOI: 10.1016/j.cagd.2005.06.006 2.

[PYK*20] PARK, JONGHA, YUN, JIHYE, KIM, NAMKUG, et al. "Fully Automated Lung Lobe Segmentation in Volumetric Chest CT with 3D U-Net: Validation with Intra- and Extra-Datasets". *Journal of Digital Imaging* 33.1 (2020), 221–230. DOI: 10.1007/s10278-019-00223-1 1.

[SC*19] SHARP, NICHOLAS, CRANE, KEENAN, et al. *Geometry Central*. 2019. URL: https://geometry-central.net/ 6.

[SC20] SHARP, NICHOLAS and CRANE, KEENAN. "You Can Find Geodesic Paths in Triangle Meshes by Just Flipping Edges". *ACM Transactions on Graphics* 39.6 (2020), 1–15. DOI: 10.1145/3414685.3417839 4, 6, 8, 10.

[Sha08] SHAMIR, ARIEL. "A Survey on Mesh Segmentation Techniques". *Computer Graphics Forum* 27.6 (2008), 1539–1556. DOI: 10.1111/j.1467-8659.2007.01103.x 3.

[SSC19] SHARP, NICHOLAS, SOLIMAN, YOUSUF, and CRANE, KEENAN. "Navigating Intrinsic Triangulations". *ACM Transactions on Graphics* 38.4 (2019), 1–16. DOI: 10.1145/3306346.3322979 6.

[XJZ*23] XU, RONGYAN, JIN, YAO, ZHANG, HUAXIONG, et al. "A Variational Approach for Feature-Aware B-spline Curve Design on Surface Meshes". *The Visual Computer* (2023). DOI: 10.1007/s00371-023-03001-x 2.

[ZGSZ03] ZACHOW, STEFAN, GLADILIN, EVGENY, SADER, ROBERT, and ZEILHOFER, HANS-FLORIAN. "Draw and Cut: Intuitive 3D Osteotomy Planning on Polygonal Bone Models". *International Congress Series* 1256 (2003), 362–369. DOI: 10.1016/S0531-5131(03)00272-3 1.