Pacific Graphics 2014
J. Keyser, Y. J. Kim, and P. Wonka
(Guest Editors)

Volume 33 (2014), Number 7

Anisotropic geodesics for live-wire mesh segmentation

Yixin Zhuang[1,2] and Ming Zou[2] and Nathan Carr[3] and Tao Ju[2]

[1]HPCL, National University of Defense Technology, China
[2]Washington University in St. Louis, USA
[3]Adobe, USA

Abstract

We present an interactive method for mesh segmentation that is inspired by the classical live-wire interaction for image segmentation. The core contribution of the work is the definition and computation of wires on surfaces that are likely to lie at segment boundaries. We define wires as geodesics in a new tensor-based anisotropic metric, which improves upon previous metrics in stability and feature-awareness. We further introduce a simple but effective mesh embedding approach that allows geodesic paths in an anisotropic path to be computed efficiently using existing algorithms designed for Euclidean geodesics. Our tool is particularly suited for delineating segmentation boundaries that are aligned with features or curvature directions, and we demonstrate its use in creating artist-guided segmentations.

1. Introduction

Mesh segmentation is a fundamental task that has numerous graphics applications including shape matching, parameterization, texturing, and remeshing. While extensive research has been conducted on automatic algorithms [APP*07, Sha08], computing semantically meaningful segmentations remains an unsolved problem. In addition, automated methods lack convenient controls by the users, which makes them less suited in human-centered application domains such as creative design. As a result, semi-automatic tools have been developed that allow users to actively participate in the segmentation process [ScH00, And00, CGF09, HL09, LL02, LLS*05, FKS*04, TPSHSH13]. The ultimate goal of interactive segmentation is to give users sufficient control over the results while minimizing their effort.

1.1. Motivation

Our work is inspired by the *live-wire* metaphor for interactive image segmentation [MB95]. Given a seed location provided by the user, the computer dynamically displays a curve segment (the "wire") connecting the seed and the current cursor location as the user moves the cursor over the image (hence "live"). The wire is computed to "snap" onto nearby image features. Once the user is satisfied with the wire, she fixes it by a mouse click, and the current cursor location is used as the seed point for the next wire.

The live-wire interaction offers a good trade-off between controllability and ease-of-use for segmentation. The user has direct control over the segmentation boundaries - a wire is fixed only when the user is happy with it. This is in contrast to other approaches, such as snakes and level-sets, where the user cannot directly control the exact shape of the boundaries. On the other hand, compared with purely manual drawing of boundaries, live-wires make it is much easier and more accurate to trace boundaries that lie at strong image features - they can be captured by long wires with few clicks. More inputs (and shorter wires) are needed only on more homogeneous parts of the image, where the user may exercise their judgement or creativity.

1.2. Problem statement

To extend the live-wire interaction for mesh segmentation, the key research problem is to define a wire between two points on the surface that forms a natural segment boundary. Applications and perceptual studies point to several line types that are candidates of "good" segment boundaries:

- **Features lines:** Both perception literature [HR83] and user studies [CGF09] suggest that humans partition objects into parts along strong concavities (i.e., valleys). On the other hand, applications like texturing require each textured patch to have low curvature variation, implying that both concavities and convexities (i.e., ridges) should be captured by the patch boundaries.

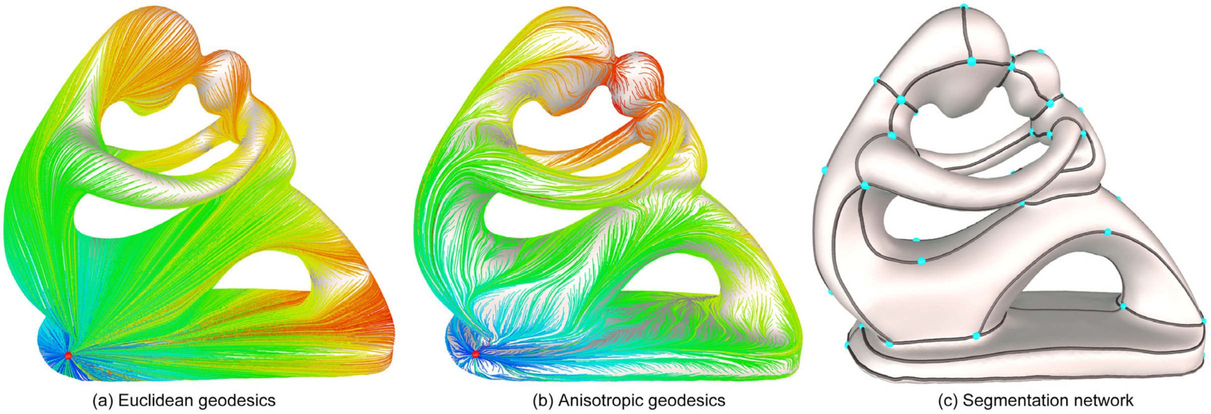

Figure 1: *Shortest paths, colored by their lengths (blue: short; red: long), in the Euclidean metric (a) and our anisotropic metric (b) from a single vertex (red) in the Fertility model, and a live-wire network where each wire (black) is a geodesic in our metric between two seeds (blue) (c).*

- **Curvature lines:** In conceptual design, artists often convey a 3D shape using lines that follow the principle curvature directions [BWSS12]. Similar properties were found on lines drawn by artists to depict an existing surface [CGL*08].
- **Shortest paths (geodesics):** The minimality rule from perception [HR83] suggests that humans tend to decompose a shape along locally-shortest boundaries.

Definitions and algorithms for these line types, while available, are quite distinct from each other. Furthermore, with the exception of geodesics, the other line types cannot be defined between two arbitrary points on the surface. The challenge we will address in this work is to define, and compute, wires that capture the essence of these candidate lines as much as possible while constrained at the ends.

1.3. Approach

To address the challenge, we define a wire as the shortest path in a non-Euclidean, *anisotropic* metric over the surface. The metric is designed so that paths that are closer to salient features or aligning better with curvature directions are shorter. At a vertex, the metric prefers paths that are better aligned with the maximal (or minimal) curvature direction. This preference is more pronounced at vertices where the maximal and minimal curvatures are more differentiated, which tend to be near prominent ridges and valleys.

To efficiently compute the wires, we embed the mesh in the new metric so that the anisotropic geodesics on the original mesh become Euclidean geodesics on the embedded mesh. In this way, we leverage existing efficient implementations of exact geodesics over triangular meshes. Once the user locates a seed vertex, geodesics are computed from the seed to every other vertex. As the user moves the other end vertex, the shortest path is visualized at interactive speed.

Figure 1 (a,b) compares the shortest paths in the standard Euclidean metric and in our anisotropic metric. Note that geodesics in our metric tend to follow features and curvature directions. Figure 1 (c) shows a segmentation network created using our live-wire tool.

Contributions The contribution of this work is three-fold:

1. A novel interface for interactively drawing segment boundaries on meshes. The interface closely mimics the classic live-wire metaphor for image segmentation.
2. A new tensor-based anisotropic metric on surfaces for line drawing. Compared to existing metrics, the geodesics under our metric better snap to salient features (e.g., ridges and valleys) and is also more stable in homogeneous and feature-less regions.
3. An efficient method for embedding a triangular mesh in a tensor-based anisotropic metric using local subdivision. While lacking theoretical support, we empirically observed that the method is highly effective in reducing the coverage of triangles that fail triangular inequality.

2. Background and previous work

We first discuss the mathematical background of surface metrics, then we give a brief review on previous work on defining surface metrics, computing anisotropic geodesics, and interactive mesh segmentation.

2.1. Surface metrics: background

A metric is a way to measure distances on the surface. To define a metric on a smooth surface S in R^3, we first define a scalar *norm*, $g_x(v)$, for each point $x \in S$ and a tangent vector v at x. Let $\tau : [0,1] \rightarrow S$ be a curve on the surface. The length of τ using this norm is

$$l_g(\tau) = \int_0^1 g_{\tau(t)}(\tau'(t))dt$$

The distance between two points x,y on S is the infimum of lengths among all curves between x,y. The path that realizes this distance, if exists, is a geodesic under the metric.

A norm (and the resulting metric) is isotropic if it is direction-invariant, that is $g_x(v) = c_x(\|v\|)$ for some function c_x. As a special case, the Euclidean metric, which gives rise to the standard geodesics, is defined by the Euclidean norm $g_x(v) = \|v\|$.

An anisotropic norm is often defined by associating each point $x \in S$ with a 2D symmetric tensor M_x. Expressing v as a 2D vector in a chosen coordinate frame on the tangent plane at x, the norm can be defined as

$$g_x(v) = \sqrt{v^T M_x v} \tag{1}$$

To understand the norm in a geometric way, let λ_1, λ_2 be the eigenvalues of M_x such that $\lambda_1 \geq \lambda_2$, and e_1, e_2 be the corresponding eigenvectors. If θ is the angle formed between v and e_1, then we can re-write the norm as

$$g_x(v) = \|v\|\sqrt{\lambda_1 \cos(\theta)^2 + \lambda_2 \sin(\theta)^2} \tag{2}$$

From this formulation, it can be seen that $g_x(v)$ changes from $\sqrt{\lambda_1}\|v\|$ (greater) to $\sqrt{\lambda_2}\|v\|$ (smaller) as the direction of v sweeps from e_1 to e_2 (see Figure 2). The norm effectively encourages geodesics to align with e_2. The extent of this encouragement, relative to curves aligned with e_1, can be captured by the ratio $\sqrt{\lambda_1/\lambda_2}$, which we call the *anisotropy* of the norm at x. The *maximum anisotropy* of a metric is the supremum of $\sqrt{\lambda_1/\lambda_2}$ over all points on S.

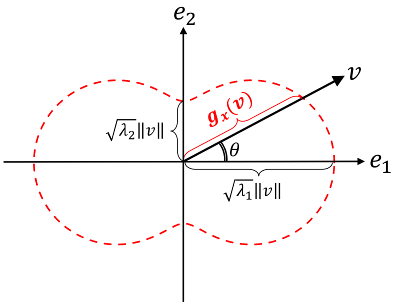

Figure 2: *A tensor-based anisotropic norm.*

In this work, we consider anisotropic norms that are aligned with principle curvatures. In the following, we let κ_1, κ_2 denote the maximum and minimum curvature at a surface point, such that $\|\kappa_1\| \geq \|\kappa_2\|$, and u_1, u_2 be the corresponding curvature directions.

2.2. Defining metrics

While Euclidean geodesics have been commonly used in interactive drawing [ScH00, And00, CGF09], they usually do not follow salient features, in particular ridges, or curvature directions (see Figure 1 (a)). Several isotropic metrics were proposed for terrain navigation [LMS96] and defining feature curves on surfaces [KMG98, HL09]. However, since an isotropic norm does not consider the direction of the path, it cannot explicitly enforce the geodesics to align with principle curvature directions.

A number of tensor-based anisotropic metrics have been proposed in the context of quad meshing and segmentation [CBK12, PSH*04, KMZ10]. However, when the eigenvectors of these tensors (e_1, e_2) are aligned with the principle curvature directions (u_1, u_2), the geodesics under these metrics either become unstable in feature-less regions or may fail to snap to salient features. More specifically,

- **Campen's metric:** Campen et al. [CBK12] used a norm with a constant anisotropy by setting
$$\lambda_1 = \alpha, \quad \lambda_2 = 1 \tag{3}$$
for some global constant $\alpha > 1$. Note that M_x is discontinuous at an umbilical or saddle point, where curvature directions u_1, u_2 are not well defined. As a result, the metric and the resulting geodesics become rather unstable in flat, spherical or saddle-like surfaces (see Figure 3 left).
- **Pottmann-Kovacs metric:** Pottmann et al. [PSH*04] and Kovacs et al. [KMZ10] used a norm with a variable anisotropy,
$$\lambda_1 = \kappa_1^2 + \beta, \quad \lambda_2 = \kappa_2^2 + \beta \tag{4}$$
for some choice of β. The norm is well-defined at an umbilical or saddle point, where $g_x(v)$ becomes an isotropic norm $\|v\|\sqrt{\kappa_1^2 + \beta}$. On the other hand, since the norm scales with the curvature magnitude, geodesics tend to avoid highly-curved parts of the surface even if features are present. This is demonstrated in the Rocker arm example of Figure 4, where geodesics in this metric (green) cannot follow the curved ridges and valleys due to the presence of a nearby flat region.

Non-tensor based anisotropic metrics have also been studied for anisotropic heat diffusion [WHSQ11, HLS*13], quad meshing [TPP*11], and computing curvature-aware geodesic maps [YSS*12]. These metrics are either not directly related to principle curvature directions [WHSQ11, HLS*13] or have similar limitations as the tensor-based metrics discussed above. For example, the metric of Tarini et al. [TPP*11], like Campen's metric, may become unstable near the singularity of the curvature field. The geodesics under the metrics of Yoo et al. [YSS*12], like Pottmann-Kovacs metric, tend to avoid directions of high curvature or high curvature variation even if the direction is aligned with a feature (e.g., a highly curved ridge).

2.3. Computing anisotropic geodesics

Methods that compute geodesics in an anisotropic metric generally take one of the two approaches. We will briefly

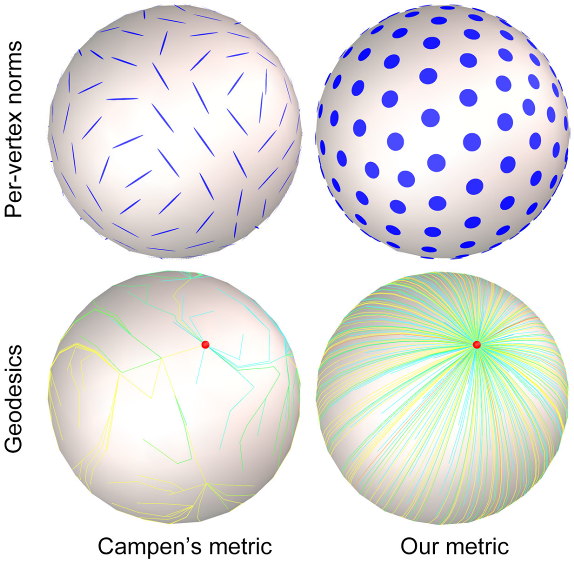

Figure 3: *Comparing Campen's metric (left) with our metric (right) on a tessellated sphere, showing the norm at each vertex as an ellipse with λ_1, λ_2 as its major and minor axes (top) and single-source all-destination geodesics (bottom). Note that Campen's metric is unstable on this near-umbilical surface. We use $\alpha = 20$ in Campen's metric and $\gamma = 0.1$ in our metric.*

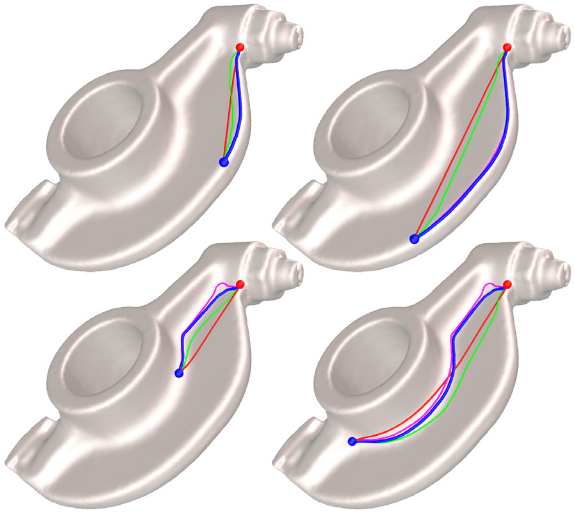

Figure 4: *Comparing geodesics in the Euclidean (red), Campen's (magenta), Pottmann-Kovacs (green) and our (blue) metrics when endpoints lie on a curved ridge (top) and a curved valley (bottom). Note that Pottmann-Kovacs geodesics tend to avoid the features when flat regions are nearby.*

review methods in each strategy, with an eye towards their balance between efficiency and accuracy. Note that while many of these methods return only a distance field, smooth geodesic paths can be traced from the gradient of the field as in [YSS*12].

Direct computation: The simplest way to approximate the anisotropic geodesics is computing the shortest path in the edge graph of the mesh, where each edge is weighted by its length in the new metric [HL09]. However, the restriction to edges results in poor approximation to the geodesics. While more accuracy can be gained by adding Steiner points and edges [LMS99], the addition comes at the cost of significantly increased computational overhead [CHK13]. Similar performance issue can be found with the ordered upwind method (OUM) [SV03, YSS*12], which extends the Fast Marching method [KS98] to handle general anisotropy but at much higher computational cost.

Several recent methods trade off accuracy of the distance field for efficiency. Campen et al. [CHK13] proposed the Short-Term Vector Dijkstra (STVD) method that approximates the geodesics by a combination of triangle edges and Euclidean shortest paths. It was shown that a good balance between the accuracy and efficiency can be achieved by a suitable choice of the parameter (k). More recently, de Goes et al. [dGLB*14] proposed to compute anisotropic geodesic distances by adapting the efficient heat diffusion method [CWW13] using an generalized Laplacian operator. While no evaluation of the accuracy was provided in the paper, experiments using a similar diffusion approach [CHK13] suggested possible accuracy issues when the metric has moderate-to-high anisotropy. In addition, parameter tuning is required in both STVD and the diffusion method.

Mesh embedding: Alternatively, one may first find an embedding S' of the input surface S, so that anisotropic geodesics on the S map to Euclidean geodesics on S'. A discrete embedding can be found by replacing each edge length in S by the distance evaluated in the new metric.

The advantage of mesh embedding is that it transforms the anisotropic geodesic problem on S into the standard Euclidean geodesic problem on S', for which accurate and efficient solutions exist. The main challenge here is that the new edge lengths may not preserve triangle inequality, a requirement by all Euclidean geodesic algorithms. Previous works have found that *invalid* triangles, which fail triangle inequality, are common when the metric has large anisotropy [KMZ10, CHK13]. Campen et al. [CHK13] restores triangle inequality by solving a least-square optimization problem with inequality constraints. However, the optimization process can be time-consuming; embedding a mesh with 200K faces could take almost a minute (see Figure 8). Also, fixing the invalid triangles would involve changing the edge lengths in other triangles.

We introduce a much more efficient strategy for ensuring triangle inequality based on local subdivision, which reduces the running time to a fraction of a second (Section 4). Even when used with the exact Euclidean geodesic algorithm [SSK*05], our method achieves sufficient interactivity for live-wire segmentation. Compared with direct computation methods with comparable efficiency (e.g., STVD and the diffusion method), our algorithm is parameter-free and simple to implement.

2.4. Interactive surface segmentation

Various types of interaction modes have been proposed for segmentation or line-drawing on surfaces, each striking a trade-off between controllability and ease-of-use. At one end of the spectrum, the user can draw the entire boundary network by sketching [TPSHSH13]. Although the user has full control over the segmentation, tracing features is tedious and error-prone. Some methods allow users to give imprecise input, such as an initial segment boundary [LL02] and a cutting plane [LLS*05], or partial input, such as a stroke [FKS*04]. This input is then optimized or completed by the system, for example using shortest paths [FKS*04] or geometric snake [LL02,LLS*05]. The user has even less control over the shape of the lines in these modes.

The live-wire interaction is most similar to the *click-and-connect* paradigm [ScH00,And00,CGF09,HL09], where the user clicks on points (seeds) sequentially and lines are filled in automatically between successive seeds. While the user still has complete control over the line shape, this is done in trial-and-error: if the filled line does not have the desired shape, the user would need to click on different seeds or add more seeds. Live-wire interaction reduces this exploration time by dynamically updating the wire as the user moves the cursor without any clicking.

3. An anisotropic metric

Our goal is to design a metric g on the surface such that geodesics in this metric are likely to be segment boundaries. As discussed before, this means that the geodesics should snap to nearby ridge or valley features, if present, and strive to follow principal curvature directions. Our norm uses a tensor M_x that is aligned with the local curvature directions.The eigenvalues of M_x are specifically designed to avoid the drawbacks of previous metrics [PSH*04, KMZ10, CBK12].

To avoid instability near umbilical or saddle points, our tensors have a variable anisotropy $\sqrt{\lambda_1/\lambda_2}$ that reduces to 1 when $\|\kappa_1\| = \|\kappa_2\|$. To remove dependency of the norm on the magnitude of curvature, we instead scale the norm by the *difference* in curvature values $s_x = \|\kappa_1\| - \|\kappa_2\|$. We have found that s_x is a good indicator of feature strength. It is zero at an umbilical, planar or saddle point. Higher values of s_x imply a more significant curvature variation in one principal curvature direction over the other. In our experiments we observed that local maxima of s_x tend to co-locate with ridges and valleys, while the magnitude of s_x captures the visual importance of the features (see Figure 5).

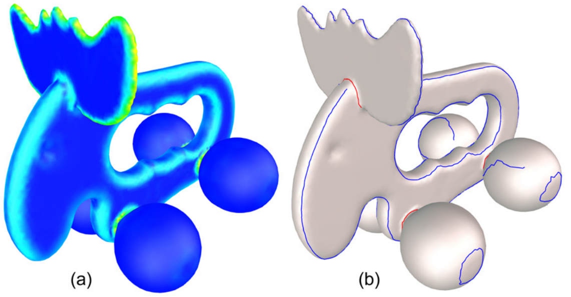

Figure 5: *Heat color plot of curvature difference s_x on Elk (a) and ridge and valleys computed by [YBS05] (b). Note that local maxima of s_x are aligned with features, and the magnitude of s_x corresponds well to visual prominence of these features.*

Specifically, the eigenvalues of both of our tensors have the following form,

$$\lambda_1 = 1 + \gamma s_x, \quad \lambda_2 = \frac{1}{1+\gamma s_x}, \tag{5}$$

where $\gamma \geq 0$ is a user-given global parameter. A smaller γ results in a norm closer to the Euclidean norm. The anisotropy of the norm reduces to 1 when $s_x = 0$. Note that since s_x is *not* scale-invariant, the metric changes when the model is scaled even with the same γ. In practice, we scale the input model to fit within a unit cube before computing the metric. We used $\gamma = 0.1$ in all our tests unless otherwise stated.

To be able to trace curves that are aligned with either minimal or maximal curvature directions, we define two tensors, M_x^{min}, M_x^{max}, both sharing the same set of eigenvalues as defined above. The minor eigenvector of M_x^{min} (resp. M_x^{max}) is aligned with the minimal (resp. maximal) curvature direction at x. In our segmentation interface, the geodesics in both metrics are displayed for the user to select (see Section 5).

We compare our M_x^{min} metric with the Euclidean metric and previous anisotropic metrics in Figure 6. The discrete curvature is computed using [Rus04], and the computation of the geodesics will be described in the next section. Observe that our metric is more stable than Campen's metric. Also, our norms are shaped in a way that paths aligned with the minimal curvature direction at strong features are shorter than paths over a flat region (see the inserts). As a result, geodesics in our metric can better follow the features than those in Pottmann-Kovacs metric. More comparisons can be found in Figures 3 and 4.

4. Computing anisotropic geodesics

Following the mesh embedding approach for computing anisotropic geodesics [CHK13], we propose an alternative method for removing invalid triangles (those that fail triangle inequality) on the discretely embedded mesh. We end the

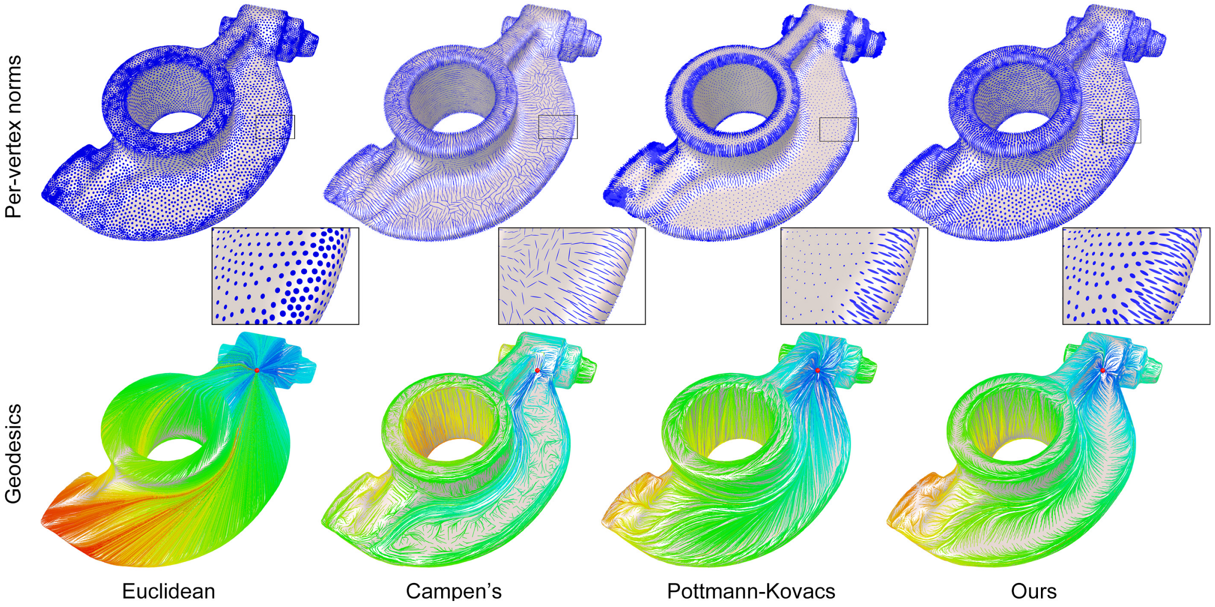

Figure 6: *Comparing the Euclidean metric (left), Campen's metric (mid-right), Pottmann-Kovac metric (mid-left) and our metric (right) on the rocker arm model, showing per-vertex norms (top) and single-source all-destination geodesics (bottom). Note that our metric results in more stable and better feature-following geodesics. To make a fair comparison, the parameter in each anisotropic metric is chosen to achieve the same maximum anisotropy of 20.*

section with a discussion on computing the geodesics after embedding.

4.1. Local subdivision

To motivate our approach, consider the ideal scenario where the three vertices i, j, k of a triangle share the same tangent plane as well as the tensor that defines the norms g_i, g_j, g_k. That is, the lengths of the three edges are modified by the same norm. On the other hand, observe from Equation 2 that any tensor-based norm $g_x(v)$ corresponds to a linear transformation (scaling in the direction of e_i by the amount λ_i for $i = 1,2$), which preserves triangle inequality. While such ideal case rarely happens, we can argue that the triangle inequality is more likely to hold when the normal directions and norms at the triangle vertices are similar.

Our approach is to subdivide triangles so that the normal directions and norms at the vertices in a subdivided triangle become more similar. While many subdivision schemes can be considered, we find the simple scheme of bisecting an invalid triangle along its longest edge works well. Specifically, we pick an invalid triangle, create a new vertex at the midpoint of its longest edge (in the new metric), and bisects both triangles that share this edge. To obtain the norm tensor at a new vertex p on an edge with end vertices i, j, we first compute p's normal vector by linear interpolation of the normals at vertices i, j, then linearly interpolate the norm tensors of i, j after they are rotated onto the tangent plane of p. The norm at p is then used to evaluate the lengths of its incident edges. The process is repeated until there is no more invalid triangles.

When using our anisotropic metric (Equation 5), triangle subdivision was able to eliminate all invalid triangles on all our test models. As shown in Figure 7, geodesics on the embedded surface created by our subdivision method are qualitatively similar to those created by the global optimization method of [CHK13]. On the other hand, subdivision achieves a significant speed-up over global optimization, as reported in Figure 8. Note that while the time used by optimization only scales with the model size, the time used by triangle subdivision scales with both the model size and the maximum anisotropy.

When the anisotropic metric is highly variant, our method may run for excessive number of iterations and sometimes even fail to terminate. Nevertheless, we observe that the remaining invalid triangles always become smaller in size and more isolated as subdivision goes on. An example using Campen's metric is shown in Figure 9. While a theoretical analysis is lacking, we hypothesize that subdivision has the effect of diminishing invalid triangles to isolated points. The number and locations of these points may depend on both the smoothness of the metric and the triangle shape of the mesh.

Since the remaining invalid triangles tend to be very small and isolated, they have a negligible contribution to the geodesics between the original mesh vertices. Hence one may adopt the following approach to handle an arbitrary

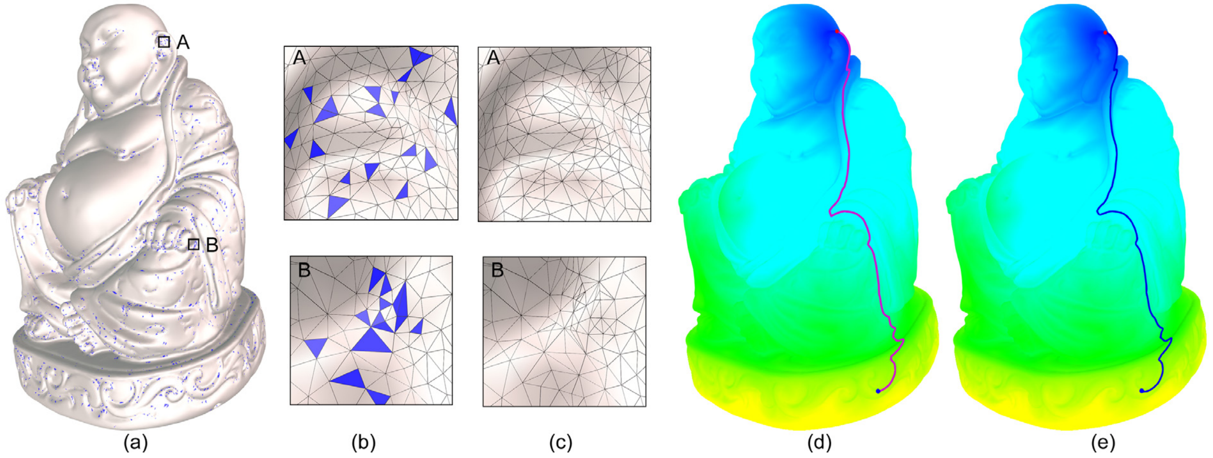

Figure 7: *The Buddha model with invalid triangles in blue (a), inserts showing triangles before (b) and after (c) subdivision. No invalid triangles exist on the subdivided model. Shortest path computed on the surface embedded using global optimization [CHK13] (d) and our triangle subdivision (e) are compared, as well as the geodesic distances (plotted in heat color).*

	Max Anisotropy	Computing Norms	Embedding			Computing Wires	
			Opt	Sub	# Triangles (invalid/added)	Seeding	Exploring
Rocker (50k)	10	0.065	10.72	0.003	45/100	0.43	0.001
	20	0.065	11	0.008	420/1088	0.45	0.001
	30	0.065	10.93	0.019	1144/3102	0.45	0.001
Feline (100k)	10	0.149	22.49	0.007	135/306	1.080	0.001
	20	0.149	22.68	0.016	760/1742	1.081	0.001
	30	0.150	23.55	0.031	1890/4632	1.001	0.001
Buddha (200k)	10	0.305	51.71	0.019	648/1380	2.003	0.001
	20	0.305	51.62	0.050	2866/6440	2.138	0.001
	30	0.305	52.18	0.098	6157/14850	1.990	0.001

Figure 8: *Running time (in seconds) for computing per-vertex norms, for embedding by global optimization [CHK13] (Opt) and by our triangle subdivision (Sub), and for computing wires on three models with varying sizes using our anisotropic metric where γ is chosen to achieve different maximum anisotropy. The timings were obtained on a 3.5GHz PC with 4GB of memory.*

metric in practice. The invalid triangles can be prioritized by their perimeter (larger triangles are subdivided first) and the subdivision is stopped once all remaining invalid triangles have perimeters smaller than a user-given threshold σ. The remaining invalid triangles will be simply treated as "holes" when computing the geodesics. All examples in this paper using Campen's and Pottmann-Kovacs metrics were generated using $\sigma = 0.001$ (assuming the model has been scaled to fit in a unit cube, as discussed in Section 3). We always let subdivision run to completion when using our own metric in the segmentation tool.

4.2. Computing geodesics

Given a mesh S and a metric, embedding only needs to be computed once, resulting in a refined mesh S' (for which only edge lengths are known). During live-wire interaction, given two end vertices of S, Euclidean geodesics between the corresponding vertices on S' need to be computed in real-time and mapped back to the original mesh S.

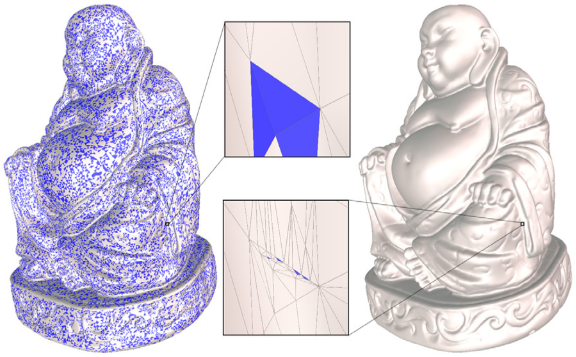

Figure 9: *Invalid triangles under Campen's metric with maximum anisotropy 20 on the original mesh (left) and after subdivision (right).*

We distinguish two types of actions in live-wire interaction, the *seeding* action where the user clicks to fix one end of the wire, and the *exploring* action where the user moves the other end of the wire with the cursor. It is most important that the user can explore the wires with as little delay as possible, whereas a small delay may be tolerable after seeding. As a result, we do more computation after seeding to gather enough information so that geodesics to individual vertices can be traced quickly.

With these requirements in mind, we adopted the classical MMP algorithm [MMP87] as implemented in [SSK*05], modified to rely only on edge length information rather than vertex coordinates. We observed in our tests that the seeding time usually takes at most one or two seconds, while tracing is done instantly (see Figure 8). While this is sufficiently fast for the size of models in our test, additional improvement in speed would be desirable for larger models, and could be achieved using approximate algorithms such as [YWH13, KS98, CWW13].

5. Interface

To start a drawing session, the user selects a seed vertex by a mouse click with Ctrl key down. As she moves the mouse cursor over the mesh, our tool displays two wires corresponding to the two metrics defined by tensors M_x^{min}, M_x^{max}, both connecting the seed to the vertex under the cursor location. The wire with a shorter length in the respective metric is *active* (shown in solid line), but she can freely toggle between the two wires using the mouse scroll wheel. A mouse click places a new seed at the cursor location, which adds the active wire to the curve network. She can terminate the current sequence of curves with a right-mouse-button click.

Occasionally the wire does not have the shape that the user desires. The surface may be missing features or distinct curvature directions, or she may wish to exercise creativity. In such a case, we allow the user to draw a (possibly incomplete) scribble on the surface indicating the shape of the desired line. She does this by clicking-and-dragging the mouse cursor over the mesh while holding the Shift key down. The scribble is recorded as a polygonal path on the mesh. We then change the metric so that future wires in the vicinity will follow the scribbled path.

To achieve this effect, we alter the norms at the vertices along the scribbled path so that 1) the minor eigenvector e_2 of the tensor in the active metric are aligned with the scribble direction, and 2) the two eigenvalues of the tensor are set to be $\lambda_1 = \eta, \lambda_2 = 1/\eta$ for some constant anisotropy η (we use 20). These norms are then propagated to vertices further away from the scribbled path and blended with existing norms there in a weighted manner, so that the contribution from the propagated norms drops to zero outside the k-ring neighborhood of the scribble path (we use $k = 10$). As the metric is modified, the mesh needs to be re-embedded. Thanks to our efficient embedding method using triangle subdivision (Section 4.2), each scribble can be performed with only one or two seconds of delay.

An example is shown in Figure 10, where the user creates a rounded wire on the flat surface of the Elk's head using a scribble. Note that the scribble does not need to be a complete curve, and hence it is easier to create than in a purely manual drawing tool.

6. Results

Figure 11 illustrates the importance of humans in the segmentation process. The first two columns show results of automatic algorithms for computing feature lines [YBS05] and segmentations (VSA of [CSAD04] and Exoskeleton of [dGGDV11]). The third column shows a curve network created by an artist using a free-hand drawing interface, where he draws on the screen and the drawing is projected onto the surface. Note that the human-created curves can be quite different from automatic solutions, particularly in regions missing strong features.

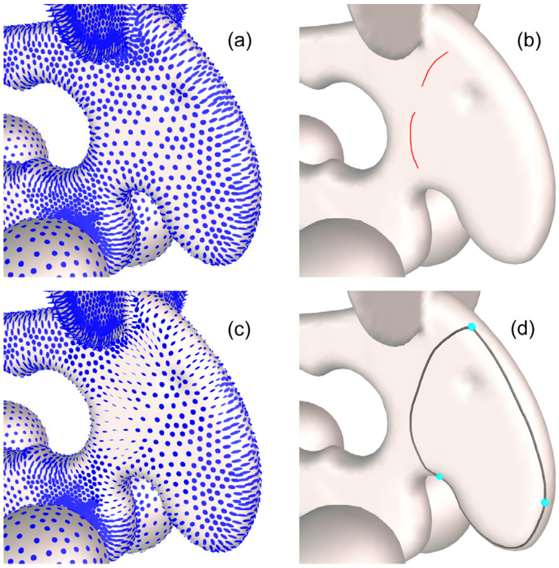

Figure 10: *Initial metric on the Elk (a), a user scribble (b), modified metric based on the scribble(c), and wires in the new metric (d).*

We used our live-wire tool to create curve networks that mimic the artist's sketches, shown in the last column of Figure 11. We found that the majority of the artist's curves can be well captured by the wires, and a small number of seeds (blue dots) are needed. Scribbles are used only in a few places (marked by arrows) where the artist's intention differs from local curvature directions. Our tool makes it particularly easy to trace sharp features (e.g., outer rim of the Rocker arm and the rim of the Feline's wings), which is difficult and tedious with free-hand drawing.

Additional results of our tool are shown in Figure 12. The interaction time in a typical segmentation session is between 5 and 10 minutes. Please refer to the accompanying video for a demonstration of our tool.

7. Conclusion

We present an interactive tool for mesh segmentation that mimics the live-wire interaction in image segmentation. The wire is defined as the geodesics in a novel feature-aware and curvature-following anisotropic metric. To efficiently compute the wires, we propose a fast local subdivision method for embedding the mesh in the metric so that anisotropic

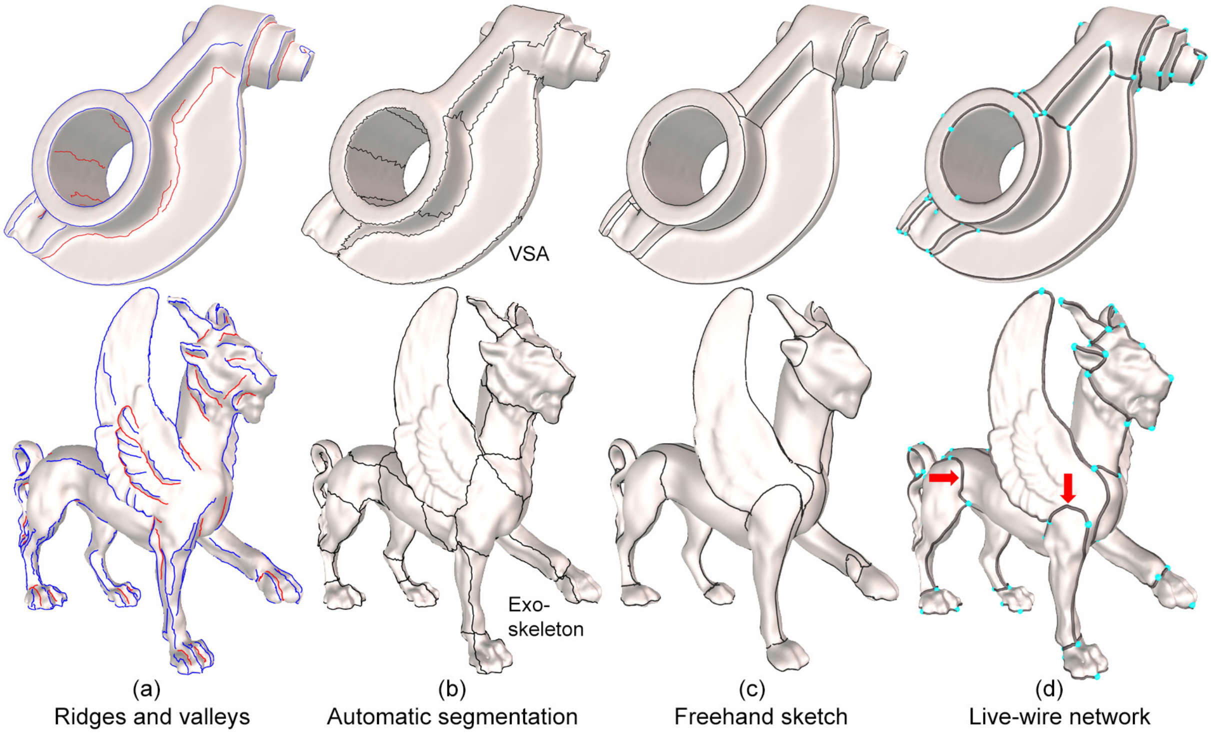

Figure 11: *Ridges and valleys [YBS05] (a), automatic segmentations (b), curves sketched manually by an artist (c), and a segmentation created using our live-wire tool guided by the artist's sketch (d). Arrows in (d) point to wires defined by scribbles. Note the difference between algorithmic curves and human-drawn curves, and the sparsity of nodes in our curve network.*

geodesics can be computed using existing algorithms for Euclidean geodesics.

This work opens the door to several interesting venue of research. On the theoretical side, our simple subdivision algorithm for mesh embedding begs more thorough investigation and analysis. Although the experimental results are promising, a more elaborate subdivision scheme (e.g., one that considers both the triangle shape and local metric variation) might have been more effective in eliminating invalid triangles and/or equipped with better theoretical guarantees such as termination and bounded approximation error. On the practical side, it would be interesting to explore anisotropic metrics whose geodesics capture other line types, such as planar curves and view-dependent lines (e.g., silhouettes), which are useful for line drawing and shape representation. One could also explore other applications of our geodesics such as feature detection [HLS*13, YSS*12] and quad meshing [TPP*11, CBK12].

Acknowledgements

We thank Daichi Ito for providing the manual segmentations in Figure 11, and authors of papers [YBS05, CSAD04, dGGDV11] for providing code and/or data. This work is supported in part by NSF grants (IIS-0846072, IIS-1302200, IIS-1319573), an NSF China grant (61379103), and a gift from Adobe.

References

[And00] ANDREWS S.: *Interactive generation of feature contours on surfaces: a minimal paths approach*. Master's thesis, University of Toronto, 2000.

[APP*07] AGATHOS A., PRATIKAKIS I., PERANTONIS S., SAPIDIS N., AZARIADIS P.: 3D mesh segmentation methodologies for CAD applications. *Computer-Aided Design & Applications 4*, 6 (2007), 827–841.

[BWSS12] BESSMELTSEV M., WANG C., SHEFFER A., SINGH K.: Design-driven quadrangulation of closed 3d curves. *ACM Trans. Graph. 31*, 6 (Nov. 2012), 178:1–178:11.

[CBK12] CAMPEN M., BOMMES D., KOBBELT L.: Dual loops meshing: Quality quad layouts on manifolds. *ACM Trans. Graph. 31*, 4 (July 2012), 110:1–110:11.

[CGF09] CHEN X., GOLOVINSKIY A., FUNKHOUSER T.: A benchmark for 3d mesh segmentation. *ACM Trans. Graph. 28*, 3 (July 2009), 73:1–73:12.

[CGL*08] COLE F., GOLOVINSKIY A., LIMPAECHER A., BARROS H. S., FINKELSTEIN A., FUNKHOUSER T., RUSINKIEWICZ S.: Where do people draw lines? *ACM Trans. Graph. 27*, 3 (Aug. 2008), 88:1–88:11.

[CHK13] CAMPEN M., HEISTERMANN M., KOBBELT L.: Practical anisotropic geodesy. *Comput. Graph. Forum 32*, 5 (2013), 63–71.

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04, pp. 905–914.

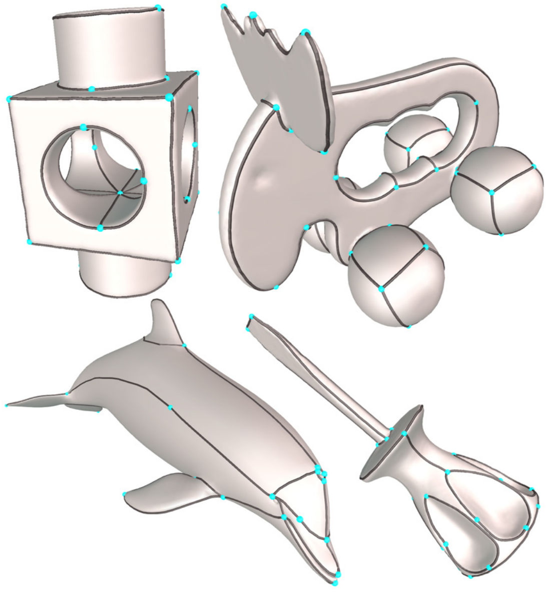

Figure 12: *More segmentation results.*

[CWW13] CRANE K., WEISCHEDEL C., WARDETZKY M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Trans. Graph. 32*, 5 (Oct. 2013), 152:1–152:11.

[dGGDV11] DE GOES F., GOLDENSTEIN S., DESBRUN M., VELHO L.: Exoskeleton: Curve network abstraction for 3d shapes. *Computers & Graphics 35*, 1 (2011), 112–121.

[dGLB*14] DE GOES F., LIU B., BUDNINSKIY M., TONG Y., DESBRUN M.: Discrete 2-tensor fields on triangulations. *Comput. Graph. Forum* (2014), to appear.

[FKS*04] FUNKHOUSER T., KAZHDAN M., SHILANE P., MIN P., KIEFER W., TAL A., RUSINKIEWICZ S., DOBKIN D.: Modeling by example. In *ACM SIGGRAPH 2004 Papers* (2004), SIGGRAPH '04, pp. 652–663.

[HL09] HSU S.-H., LAI J.-Y.: Extraction of geodesic and feature lines on triangular meshes. *The International Journal of Advanced Manufacturing Technology 42*, 9-10 (2009), 940–954.

[HLS*13] HOU T., LI S., SU Z., QIN H., WANG S.: Anisotropic elliptic pdes for feature classification. *IEEE Trans. Vis. Comput. Graph. 19*, 10 (2013), 1606–1618.

[HR83] HOFFMAN D., RICHARDS W.: Parts of recognition. *COGNITION 18* (1983), 65–96.

[KMG98] KHANEJA N., MILLER M. I., GRENANDER U.: Dynamic programming generation of curves on brain surfaces. *IEEE Trans. Pattern Anal. Mach. Intell. 20*, 11 (Nov. 1998), 1260–1265.

[KMZ10] KOVACS D., MYLES A., ZORIN D.: Anisotropic quadrangulation. In *Proceedings of the 14th ACM Symposium on Solid and Physical Modeling* (2010), SPM '10, pp. 137–146.

[KS98] KIMMEL R., SETHIAN J. A.: Computing geodesic paths on manifolds. In *Proc. Natl. Acad. Sci. USA* (1998), pp. 8431–8435.

[LL02] LEE Y., LEE S.: Geometric snakes for triangular meshes. *Comput. Graph. Forum 21*, 3 (2002), 229–238.

[LLS*05] LEE Y., LEE S., SHAMIR A., COHEN-OR D., SEIDEL H.-P.: Mesh scissoring with minima rule and part salience. *Computer Aided Geometric Design 22*, 5 (2005), 444–465.

[LMS96] LANTHIER M., MAHESHWARI A., SACK J.-R.: Approximating weighted shortest paths on polyhedral surfaces. In *In 6th Annual Video Review of Computational Geometry, Proc. 13th ACM Symp. Computational Geometry* (1996), ACM Press, pp. 274–283.

[LMS99] LANTHIER M., MAHESHWARI A., SACK J.-R.: Shortest anisotropic paths on terrains. In *In Proceedings of the 26th International Colloquium on Automata, Languages and Programming* (1999), Springer-Verlag, pp. 524–533.

[MB95] MORTENSEN E. N., BARRETT W. A.: Intelligent scissors for image composition. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques* (1995), SIGGRAPH '95, pp. 191–198.

[MMP87] MITCHELL J. S. B., MOUNT D. M., PAPADIMITRIOU C. H.: The discrete geodesic problem. *SIAM J. Comput. 16*, 4 (Aug. 1987), 647–668.

[PSH*04] POTTMANN H., STEINER T., HOFER M., HAIDER C., HANBURY A.: The isophotic metric and its application to feature sensitive morphology on surfaces. In *ECCV (4)* (2004), pp. 560–572.

[Rus04] RUSINKIEWICZ S.: Estimating curvatures and their derivatives on triangle meshes. In *Symposium on 3D Data Processing, Visualization, and Transmission* (Sept. 2004).

[ScH00] STALLING D., CHRISTIAN HEGE H.: Fast and intuitive generation of geometric shape transitions. *The Visual Computer 16* (2000), 241–253.

[Sha08] SHAMIR A.: A survey on mesh segmentation techniques. *Comput. Graph. Forum 27*, 6 (2008), 1539–1556.

[SSK*05] SURAZHSKY V., SURAZHSKY T., KIRSANOV D., GORTLER S. J., HOPPE H.: Fast exact and approximate geodesics on meshes. *ACM Trans. Graph. 24*, 3 (July 2005), 553–560.

[SV03] SETHIAN J. A., VLADIMIRSKY A.: Ordered upwind methods for static Hamilton-Jacobi equations: Theory and algorithms. *SIAM Journal on Numerical Analysis 41*, 1 (2003), 325–363.

[TPP*11] TARINI M., PUPPO E., PANOZZO D., PIETRONI N., CIGNONI P.: Simple quad domains for field aligned mesh parametrization. In *Proceedings of the 2011 SIGGRAPH Asia Conference* (2011), SA '11, pp. 142:1–142:12.

[TPSHSH13] TAKAYAMA K., PANOZZO D., SORKINE-HORNUNG A., SORKINE-HORNUNG O.: Sketch-based generation and editing of quad meshes. *ACM Trans. Graph. 32*, 4 (July 2013), 97:1–97:8.

[WHSQ11] WANG S., HOU T., SU Z., QIN H.: Multi-scale anisotropic heat diffusion based on normal-driven shape representation. *The Visual Computer 27*, 6-8 (2011), 429–439.

[YBS05] YOSHIZAWA S., BELYAEV A., SEIDEL H.-P.: Fast and robust detection of crest lines on meshes. In *Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling* (2005), SPM '05, pp. 227–232.

[YSS*12] YOO S. W., SEONG J.-K., SUNG M.-H., SHIN S. Y., COHEN E.: A triangulation-invariant method for anisotropic geodesic map computation on surface meshes. *IEEE Trans. Vis. Comput. Graph. 18*, 10 (2012), 1664–1677.

[YWH13] YING X., WANG X., HE Y.: Saddle vertex graph (svg): A novel solution to the discrete geodesic problem. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 170:1–170:12.