Eurographics Symposium on Geometry Processing 2009
Marc Alexa and Michael Kazhdan
(Guest Editors)

Volume 28 (2009), Number 5

Separatrix Persistence: Extraction of Salient Edges on Surfaces Using Topological Methods

T. Weinkauf[1,2] and D. Günther[1]

[1]Zuse Institute Berlin, Germany
[2]Courant Institute of Mathematical Sciences, New York University, U.S.A.

Abstract

Salient edges are perceptually prominent features of a surface. Most previous extraction schemes utilize the notion of ridges and valleys for their detection, thereby requiring curvature derivatives which are rather sensitive to noise. We introduce a novel method for salient edge extraction which does not depend on curvature derivatives. It is based on a topological analysis of the principal curvatures and salient edges of the surface are identified as parts of separatrices of the topological skeleton. Previous topological approaches obtain results including non-salient edges due to inherent properties of the underlying algorithms. We extend the profound theory by introducing the novel concept of separatrix persistence*, which is a smooth measure along a separatrix and allows to keep its most salient parts only. We compare our results with other methods for salient edge extraction.*

Categories and Subject Descriptors (according to ACM CCS): G.2.3 [Mathematics of Computing]: Discrete Mathematics—Applications; I.3.0 [Computer Graphics]: General—;

1. Introduction

Salient edges are visually prominent characteristics of a surface. Extraction techniques for these features on triangular surface meshes have become standard tools in the Graphics community and are used in a variety of applications such as e.g. measuring the quality of free-form surfaces [Hos92], registration of surfaces [Sty04], non-photorealistic rendering [IFP95], and surface segmentation [SF04].

In this paper we follow the common description of salient edges as line-type features of a surface along which the principal curvatures $\kappa_{min}, \kappa_{max}$ are locally extremal: perceptually salient convex edges are given as maximal lines of κ_{max} fulfilling $|\kappa_{max}| > |\kappa_{min}|$, and salient concave edges as minimal lines of κ_{min} fulfilling $|\kappa_{min}| > |\kappa_{max}|$. Hence, we are interested in view-independent features of the surface.

To extract them, most techniques utilize the notion of *ridge and valley lines* [Thi96, Ebe96]. This requires first and second order derivatives of the curvature function which itself already contains second order derivatives of the underlying surface. Hence, ridges and valleys need third and fourth order derivatives of the actual input mesh and therefore they are very sensitive to noise. The robust computation of these derivatives is a challenging problem. It is still an active field of research and a large number of different approaches have already been proposed for this: fitting of a smooth surface either locally [CP03, GI04, YBS05] or globally [OBS04] as well as evaluating discrete differential operators [MDSB02, CSM03, HPW05].

In this paper we want to overcome the problem of curvature derivative estimations by developing a method for the extraction of salient edges that does not require curvature derivatives at all. To do so, we extract the *topological skeletons* of the principal curvature functions, which contain lines of minimal/maximal curvature called *topological separatrices*. They are a superset of the desired salient edges. Our method is based on *discrete Morse theory* [For98, For02], which allows us to compute the topology of a discretized function without referring to its derivatives [Lew02]. Furthermore, the algorithm works completely combinatorial on a discrete domain, i.e., numerical issues do not play a role. This makes our new method a robust, less noise-sensitive scheme for extracting salient edges on triangle meshes.

Topology in general is well-understood and thorough treatments can be found in e.g. [FG82, GH86]. The concept has been applied in a variety of applications such as remeshing of triangular surfaces using quadrilaterals [DBG*06], analyzing and visualizing scalar [PCM03] and vector fields [HH89], or identification of handles and tunnels in surfaces [GW01]. The watershed algorithm [RM00] is closely related to topology and often applied in image analysis.

However, although topological concepts have become standard tools in other areas and despite the possible algorithmic advantages due to derivative-free and purely combinatorial computations, almost no approaches exist to apply this scheme for the extraction of salient edges. As we argue in section 3, this is because existing topological approaches keep some non-salient lines, and remove some obviously salient parts of the skeleton. In particular, topology lacks a concept for measuring the strength along a feature line and all previous extraction schemes ignore that the feature strength may smoothly increase along the line. For these reasons the results of previous topological extraction schemes are less satisfying than those of ridge/valley schemes.

In this paper we address these problems by introducing a novel concept to the field of topology: the *persistence of a separatrix*. This is a smooth measure of importance for every point along a separatrix, which allows us to tell apart salient feature lines from non-salient ones. Its computation is fast and completely parameter-free. To the best of our knowledge, this paper presents the first approach to a smooth extraction of salient edges using discrete topology – thereby avoiding curvature derivatives and other numerical issues inherent to ridge/valley schemes. We conduct a thorough comparison with other edge detection methods based on both definitions, i.e., ridges/valleys and topology.

The rest of the paper is organized as follows: In section 2 we give an introduction to the necessary topological terms and concepts. We introduce our new persistence measure for separatrices in section 3 and give implementation details in section 4. Results and a comparison to other methods is given in section 5. Conclusions are drawn in section 6.

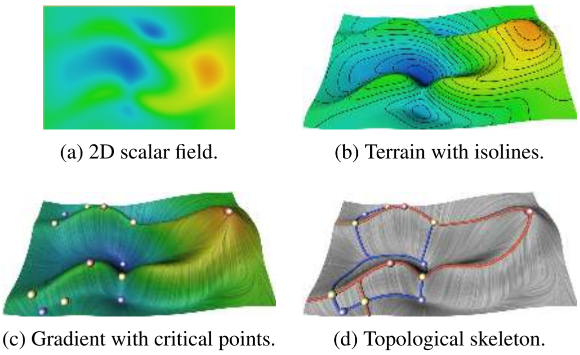
2. Theoretical Background

In the following we discuss the – for our purposes – most important terms of topology and refer the reader to the literature for more details. To ease the explanations, we consider an arbitrary 2D scalar function $f : \mathbf{R}^2 \to \mathbf{R}$, which allows us to plot its graph as a height map (see figure 1). Later, this function will be replaced by the curvature of a surface. Also note that although some of the following explanations make explicit references to derivatives, we will eventually not need them to do the actual computations.

2.1. The Topological Skeleton of a Scalar Function

Let $\mathbf{g} = \nabla f$ be the gradient of a 2D scalar function f, which is a 2D vector field pointing into the direction of steepest ascend. Considering the tangent curves (or stream lines) of $\mathbf{g}$, topology aims at describing their characteristics by segmenting the domain into regions of similar flow behavior. The key to this are *critical points*, which are isolated zeros of $\mathbf{g}$. We distinguish between (see figure 1c):

- *minimum (source)* – outflow, i.e., all tangent curves move away from this point (locally smallest value of f),
- *maximum (sink)* – inflow, i.e., all tangent curves move towards this point (point with locally largest value of f),
- *saddle* – mixture of both inflow and outflow.


(a) 2D scalar field. (b) Terrain with isolines.

(c) Gradient with critical points. (d) Topological skeleton.

Figure 1: *Topological skeleton of a simple 2D scalar function. Minima are shown in blue, maxima in red, saddles in yellow. Separatrices connecting to minima/maxima are colored in blue/red.*

While most tangent curves pass by a saddle point, four distinct tangent curves either start or end there: these are called *separatrices*, where two of them connect to minima and the other two connect to maxima. This yields a graph structure where all critical points are connected to each other by separatrices. We call this the *topological skeleton* (see figure 1d). The topological skeleton is stable under small perturbations in general: to remove critical points or to change their connectivity a perturbation is needed that exceeds the strength of these features.

The separatrices segment the domain into *topological sectors* – with the property of having only a single minimum and a single maximum in each sector. Hence, all tangent curves in a sector start at the minimum and end at the maximum. Also note, that f is monotonically increasing on a tangent curve. In other words, f behaves monotonically in each sector of the topological skeleton with respect to its gradient. The border between two adjacent sectors constitutes a change of that monotony and denotes therefore a line of minimal or maximal scalar value. The borders are given by the separatrices - hence, they are extremal lines of f.

Both concepts, separatrices and ridges/valleys, describe extremal lines of a scalar function and yield very similar results for the most dominant extremal lines. While ridges are given by a local analysis of f and its derivatives, separatrices are global structures, i.e., one cannot locally decide whether a point is part of a separatrix or not. As discussed in [SWTH07], ridges detect even rather small fluctuations of f, whereas for the topological approach a fluctuation has to be strong enough to breach the monotony of its neighborhood, i.e., by creating a local extremum. A thorough comparison of both concepts is beyond the scope of this paper; we refer the reader to [Ebe96, SWTH07, PS08].

2.2. Discrete View of Topology

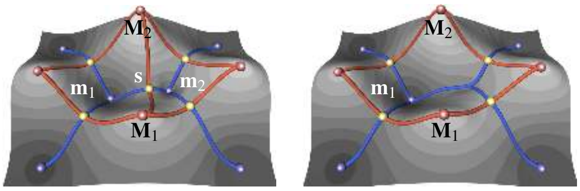
The goal of this paper is to extract salient edges of triangle meshes, i.e., discrete surfaces. It is favorable to stay in the discrete setting and handle the topology in a discrete manner as well. To do so, we choose the scheme given by Forman in his seminal work on discrete Morse theory [For98, For02]. It describes the topology of a combinatorial vector field on a cell complex. A *cell complex* of a triangle mesh consists of three classes of cells ordered according to their dimensionality: *0-cells* are the vertices of the mesh, *1-cells* the edges, and *2-cells* the triangles. Each cell is treated as an individual entity with a certain connectivity to its neighbors. A *combinatorial vector field* is now defined on a cell complex such that a n-cell points to a single incident $n+1$-cell, i.e., vertex → edge and edge → triangle. Furthermore, it is required that a cell does not point to any of its neighbors if it is already being pointed to by one of them. Critical points are cells which neither point to a neighbor nor does a neighbor point to them. Tangent curves are given as a sequence of vertex-edge links or edge-triangle links.

For our purposes, we want to encode the properties of the gradient vector field $\mathbf{g}$ in a combinatorial vector field without actually computing it. Lewiner [Lew02] and Cazal et al. [CCL03] show how this can be done using spanning trees and a watershed-like algorithm on the cell complex. Details can be found in section 4. It is important to note that the discrete gradient vector field of f can be computed without derivation – the only requirement is the ability to sort the items, i.e., a less/greater comparison on the data values suffices.

While the concept of discrete Morse theory provides a robust computational framework based on a profound theory, it also introduces an unintuitive view of separatrices: separatrices are *not* integral curves in the discrete setting, but instead a minimal line is represented as a sequence of vertex-edge links and a maximal line as edge-triangle links. The integration-free computation of separatrices contributes to the overall robustness of our approach. Consequently, separatrices are not smooth anymore and for coarse meshes the discrete nature of these features becomes obvious. However, in practice we did not encounter this problem since “real” meshes are usually dense enough.

2.3. Simplification of the Topological Skeleton

The presence of noise can cause a wealth of topological structures (often referred to as “oversegmentation”), because every local extremum causes the creation of corresponding topological structures. Topological simplification is a powerful tool to handle noise and oversegmentation. The goal is to order the structures of the skeleton according to some importance measure and successively remove all structures below a certain threshold – under the side condition that the skeleton is in a topologically consistent state after each simplification step.



(a) Before. (b) After.

Figure 2: *Cancellation of the saddle-minimum pair (*$\mathbf{s}$, $\mathbf{m}_2$*).*

In fact, there is only a single operation that maintains a consistent state: a pairwise removal (cancellation) of two critical points with opposite Poincaré index (see e.g. [FG82]). For 2D scalar fields this turns out to be either a saddle-minimum or a saddle-maximum cancellation. Let us consider a saddle-minimum cancellation starting with the skeleton of figure 2a, where the saddle $\mathbf{s}$ and the minimum $\mathbf{m}_2$ are about to be removed, whereas $\mathbf{m}_1$ and both maxima $\mathbf{M}_i$ shall remain. The separatrices connected to the maxima will be removed, and the separatrices connected to the minima will be re-used by reconnecting all separatrices of $\mathbf{m}_2$ to the remaining $\mathbf{m}_1$. The result is shown in figure 2b.

A commonly used stable importance measure for critical points is *persistence* [ELZ02, CSEH07, EHZ03]. In combination with topological simplification, it can be defined locally as the smallest deviation of f between the saddle $\mathbf{s}$ and its connected extrema $\mathbf{m}_i, \mathbf{M}_i$:

$$p_{\mathbf{s}} = \min(d_{\mathbf{s}}^{\mathbf{m}}, d_{\mathbf{s}}^{\mathbf{M}}) \tag{1}$$

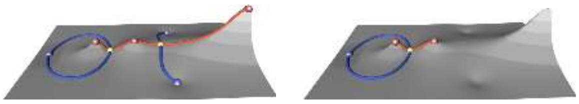
with the saddle-extremum deviations

$$d_{\mathbf{s}}^{\mathbf{m}} = f(\mathbf{s}) - \max(f(\mathbf{m}_i)) \tag{2}$$

$$d_{\mathbf{s}}^{\mathbf{M}} = \min(f(\mathbf{M}_i)) - f(\mathbf{s}). \tag{3}$$

A topological simplification is an iterative process of repeated cancellations where the saddle-extremum pair with the currently lowest persistence is canceled until a certain persistence is reached or no further cancellations are possible. Since a cancellation changes the connectivity in the graph, it also triggers an update of $p_{\mathbf{s}}$ in the neighborhood: in figure 2b, the two saddles in the right part of the domain gained persistence. This makes topological simplification an inherently global process, where the final persistence values (which match the definition in [ELZ02]) for all saddles have been determined when all (possible) cancellations have been carried out. This has proven to be a powerful tool and in the context of triangle meshes it has been applied primarily to segment surfaces into patches, see e.g. [MW99, NWB*06, BP07].

Our method is based on a persistence-based simplification of the principal curvatures $\kappa_{min}, \kappa_{max}$. However, the concept of persistence is defined for critical points only, which leads to a number of problems when one is interested in extracting salient edges as we will show in the next section.



(a) Left saddle cannot be canceled, since periodic orbits are not allowed.

(b) Right saddle is canceled and its separatrices are completely lost.

Figure 3: *Existing topological approaches keep non-salient lines while removing some salient parts of the skeleton.*

3. Salient Edge Extraction

In the following we develop our approach to salient edge extraction based on topology. First, we identify the shortcomings of existing topological methods (section 3.1). Based on this, we introduce a new concept to the field of topology: the persistence of a separatrix, which is a metric-independent importance measure for each point along an extremal line (section 3.2). We give an overview of the complete method including comments on post-processing for depiction in section 3.3.

3.1. Why Existing Topological Approaches Fail

As already discussed in the introduction, topological concepts have become standard tools for a number of applications. However, almost no approaches exist to apply this scheme for the extraction of salient edges. To the best of our knowledge, there is only a single previously existing topological method for salient edge extraction due to [SWPL08]: instead of persistence, a different importance measure for saddle points is used, which considers integrals of f over separatrices and topological sectors. Unfortunately, this does not suffice to produce results that can compete with ridge/valley schemes as we will show in section 5.2 using a real-world model.

Here, we want to identify two theoretical reasons why both the method of [SWPL08] and classic persistence canceling fail with respect to salient edge extraction. The problems arise from inherent properties of topological simplification as shown in the following.

Consider figure 3a, where the graph of a 2D scalar function is plotted as a height map. Our main focus are the maximal lines (red) and their saddle points (yellow). The left saddle has a lower persistence than the right one, so it is the first one to be canceled. However, this cancellation is topologically forbidden since it would turn the blue separatrices on the left into a so-called periodic orbit, which cannot occur in gradient vector fields. Hence, the left saddle and its *non-salient* separatrices are *enforced* to remain in the skeleton – thereby violating the idea that the skeleton contains only structures above a certain persistence threshold. In real-world models, these non-salient results are often very noticeable as false positives running through e.g. almost flat areas of the mesh.

The right saddle in figure 3a is next to be canceled. Since this is an allowed cancellation (no periodic orbit will be created), the saddle point and its separatrices are completely removed from the skeleton (figure 3b). This shows that topological simplification employs *binary decisions*: either a topological structure gets completely removed or not. Hence, the concept of topological simplification itself ignores the fact that the strength of a feature may smoothly increase along the line. This is also the case in figure 3a, where the strength of the rightmost maximal line increases while it runs upwards to the global maximum.

Summarized, existing topological approaches keep some non-salient lines, and remove some obviously salient parts of the skeleton. We address both problems by introducing the new concept of separatrix persistence in the next section.

3.2. The Persistence of a Separatrix

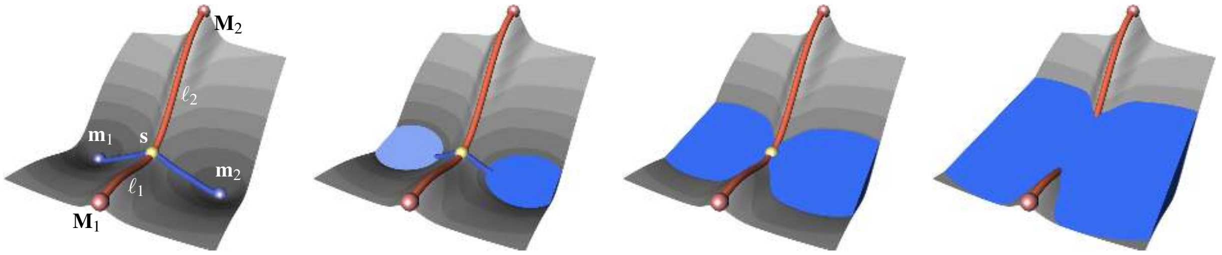
Previous schemes assigned a persistence value only to the saddles of the topological skeleton. To see why this does not suffice if we are primarily interested in extremal lines, consider the following intuitive example: figure 4a shows the topology of a simple 2D scalar field on a terrain. Imagine that rain pours onto this terrain and consequently the water assembles around the two minima as shown in figure 4b, where the blue surfaces denote the current water level. In this particular example, the water level represents the current persistence value during simplification. The two regions around the minima are still separated from each other by a “dam”, i.e., the separatrices ℓ_1, ℓ_2 (red). Water is rising and at some time it will reach the lowest point of the dam, i.e., the saddle. From this moment on, water from both regions gets mixed, i.e., the regions are merged by canceling $(\mathbf{s}, \mathbf{m}_2)$ and keeping $\mathbf{m}_1$ (figure 4c). Previous simplification schemes would also completely discard both red separatrices ℓ_1, ℓ_2 at once. However, most of the dam is still there and remains to be there even if water keeps rising (figure 4d). The two remaining parts of the dam will only be completely gone once the water level reaches their highest points, i.e., the maxima $\mathbf{M}_1, \mathbf{M}_2$.

We draw two conclusions from these observations: first, ℓ_1 and ℓ_2 have to be treated independently from each other after their saddle has been canceled. Second, the slow decay of each separatrix has to be described by an interval and not a single value. To do so, we give the following definition:

Definition 1 (Separatrix Persistence) Let $\mathbf{s}$ denote a saddle connected to the minima $\mathbf{m}_i$ and maxima $\mathbf{M}_i$ in the skeleton of f. The ***persistence of a separatrix*** ℓ is defined for each point $\mathbf{x} \in \ell$ as

$$p_\ell(\mathbf{x}) = \begin{cases} f(\mathbf{x}) - \max(f(\mathbf{m}_i)) \;, & \ell \text{ is maximal line} \\ \min(f(\mathbf{M}_i)) - f(\mathbf{x}) \;, & \ell \text{ is minimal line.} \end{cases} \tag{4}$$

Separatrix persistence measures the significance of every point on a separatrix. As it is derived from classic persistence, it inherits the stability under small perturbations. Obviously, p_ℓ reaches its highest value at the extremum. The



(a) Terrain with critical points and separatrices.

(b) Water assembles around the minima.

(c) Dam is breached at saddle.

(d) Two parts of the dam remain as water keeps rising.

Figure 4: *Separatrix persistence assigns an importance weight to each point along a separatrix. It respects that the importance of a feature may smoothly change along the line.*

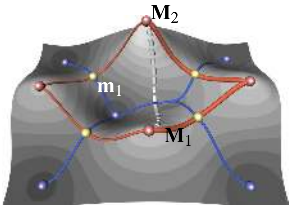


Figure 5: *The cancellation of figure 2 assigns a final persistence to the removed separatrices (gray, dashed). Neighboring separatrices gain persistence (red, thick).*

point on ℓ with the lowest persistence is the saddle $\mathbf{s}$. Note, that the value of $p_\ell(\mathbf{s})$ depends on whether ℓ is a maximal or minimal line, i.e., $p_\ell(\mathbf{s}) = d_\mathbf{s}^\mathbf{m}$ for maximal lines and $p_\ell(\mathbf{s}) = d_\mathbf{s}^\mathbf{M}$ for minimal lines following equations (2) and (3). This is very important as it allows us to treat minimal and maximal lines independently: for example, a maximal line with high persistence values can be “crossed” by a number of minimal lines with low persistence values.

During topological simplification, the persistence of affected separatrices changes. Figure 5 illustrates the effect of the saddle-minimum cancellation known from figure 2 on the persistence of the separatrices in the neighborhood of the removed minimum $\mathbf{m}_2$: since the saddles on the right side of the domain have been reconnected to the stronger minimum $\mathbf{m}_1$, their red separatrices *gain* persistence (shown as thick red lines). A similar statement holds for saddle-maximum cancellations and blue separatrices.

Note that a repeated execution of cancellations distills the most important separatrices of the topological skeleton by assigning higher persistence values to them. Therefore, we combine the computation of separatrix persistence with the process of topological simplification in order to build up a feature hierarchy, i.e., determine the most important extremal lines. To do so, we simplify the topological skeleton as described earlier and compute a final p_ℓ when separatrices are removed from the skeleton (shown as gray dashed lines in figure 5). In the case where a cancellation is topologically forbidden due to periodic orbits, we still compute a final p_ℓ for these separatrices, but leave them in the skeleton to maintain topological consistency.

The computation of the final persistence values of all separatrices is a global process since we coupled it to topological simplification. But not only from an algorithmic point of view: note that in general the distances between connected critical points increases during simplification, i.e., they are increasingly representing the non-local behavior of the scalar function. After the last cancellation, the remaining structures represent its global behavior. This is a major reason why our topological method for extracting extremal lines is more robust against noise than schemes based on the ridge/valley definition, which describes extremal lines as local features.

3.3. Method Overview and Post-Processing

The input for our salient edge extraction method is a triangulated surface mesh and its principal curvatures $\kappa_{max}, \kappa_{min}$. Our method does not depend on a specific way of computing the curvatures. In fact, we found that it works well with different schemes. For the results in section 5 we used a simple curvature estimation based on normal variation.

In order to detect edges in convex regions, we start with computing the initial topological skeleton of κ_{max}. During the following persistence-based simplification we compute the separatrix persistence as already described, i.e., with each cancellation step we determine the persistence of those separatrices that are removed from the skeleton (or enforced to remain due to periodic orbits). They are hereby fixed in the hierarchy, i.e., their importance has been determined and they are added to the output. This is done until no further cancellation is possible, but the resulting skeleton might still contain some separatrices that have not yet been added to the output. We compute their separatrix persistence and add them to the output as well. This gives us the set of all initial separatrices reconnected to longer lines during the simplification and augmented with our new separatrix persistence measure. From this set we keep only the maximal lines and parts thereof which fulfill $|\kappa_{max}| > |\kappa_{min}|$, i.e., the perceptually salient maximal lines. Similarly, we get the perceptually

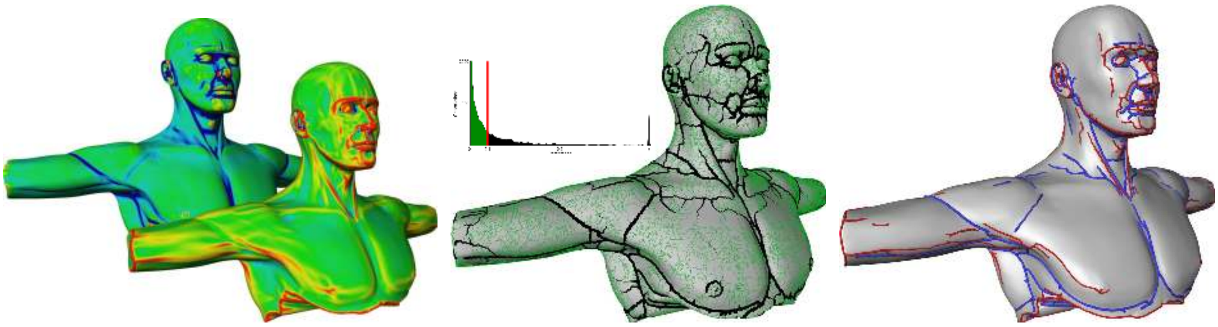


Figure 6: *Human torso. (left) $\kappa_{max}/\kappa_{min}$ of the surface shown in front/back. (middle) Reduction of the result size by removing lines with very low persistence (green); exemplified for the minimal lines of κ_{min}. Kept lines are shown in black and scaled according to separatrix persistence. (right) Perceptually salient convex edges are shown in red, concave edges in blue ($p_\ell = 20$).*

salient minimal lines from the topological skeleton of κ_{min} which fulfill $|\kappa_{min}| > |\kappa_{max}|$.

Note that this procedure is completely parameter-free, and it avoids curvature derivatives as well as other numerical problems.

The user is left with choosing an appropriate threshold for separatrix persistence in order to display the most salient edges. However, the raw result is usually rather large. There is a nice way to automatically reduce the number of output lines and confront the user with a drastically smaller result. This can be done by considering the histogram of cancellations as it is shown for the torso dataset in figure 6 (middle). It measures the number of cancellations over the persistence $p_\mathbf{s}$ of the saddles: a very high percentage of cancellations takes place at very low persistence levels indicating the amount of noise in the data set. We found that it is safe to remove all lines from the result which have been canceled at such low persistence levels. While it might be interesting to explore the possibilities of choosing a low persistence threshold automatically by e.g. analyzing the histogram, we use a fixed threshold of 0.1 in our implementation. This worked out for all our experiments and reduced the size of the result by usually more than an order of magnitude.

Finally, we allow the user to remove small lines and smooth the result in order to get rid of the high-frequent oscillations arising from the discrete nature of our features. Since separatrix persistence is a smooth measure along a line, we found it beneficial to display the lines scaled accordingly, which elucidates the strength of a feature and allows for smooth phase-outs.

4. Implementation

In the following we give some details about the algorithms for extracting the initial topological skeleton (following Lewiner [Lew02] and Cazals et al. [CCL03]) and simplifying it. Separatrix persistence is computed during the latter.

For the extraction of the initial skeleton it is assumed that the scalar field (κ_{min} or κ_{max}) is given on the vertices of the mesh. The scalar values for edges and triangles are computed as the average of the values of their incident vertices. Then the list of edges is sorted according to their value and traversed in ascending order while labels are assigned to the vertices in a watershed-like manner, i.e., for each visited edge we check if its two incident vertices already have a label:

- If not, then the vertex with the lower scalar value becomes a minimum, the other is linked to it, and both get the same label *(creation of spanning tree)*.
- If one vertex has a label, then the other gets it as well and is linked to the first *(continuation of spanning tree)*.
- If both vertices already have a label, the edge is marked to be a potential saddle *(end of spanning tree)*.

A similar procedure in descending order with edges and triangles is carried out to get the list of maxima. The final list of saddles is obtained as intersection of both sets of marked edges. Their separatrices can then easily be traced using the links set above. The complexity of this algorithm is $O(n\log(n))$ with n being the number of edges.

The skeleton is stored in a graph-like structure, which gets simplified in a combinatorial manner. The saddles are inserted into a priority queue such that the saddle with the lowest persistence comes first. Then we process the queue and cancel each saddle together with the corresponding extremum – thereby updating the neighborhood as described in section 2.3. If a cancellation is not allowed due to periodic orbits, we re-insert the saddle into the priority queue with the persistence corresponding to the opposite extrema. However, in any case we record the persistence of the separatrices that (would) have been removed from the skeleton. When the priority queue is empty, we iterate over all remaining saddles (usually a *very* low number) and compute the separatrix persistence of their separatrices as well. The complexity of this algorithm is $O(s\log(s))$ with s being the number of saddles.

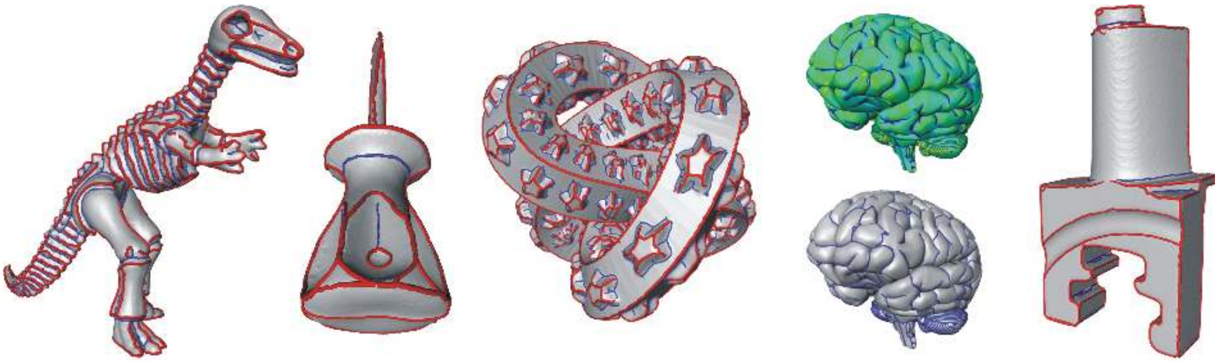


Figure 7: *Results from left to right: dinosaur, screwdriver, knot model, plot of* κ_{min} *and concave edges of the brain model, blade.*

5. Results and Comparison to Other Methods

Figure 6 exemplifies – from left to right – stages of our pipeline for extracting salient edges. Based on the principal curvatures we compute the set of salient convex and concave edges. The strength of these features is given by separatrix persistence, which is encoded for every point along the feature line. Since the result is usually rather large, we apply an automatic filtering which removes all lines that have been canceled early during topological simplification. Finally, we choose a threshold for separatrix persistence to depict the most important features. Convex edges are typically displayed in red, concave edges in blue. All models in this paper have been processed this way. Figure 7 shows further results.

We already commented on the complexity of our algorithms in section 4. While the time for the extraction of the initial skeleton depends on the size of the input mesh, the time for the topological simplification is coupled to the complexity of the skeleton. In all practical cases we found that the overall computation time is equally distributed between both stages. Our technique is as fast as the other tested methods: the feline data set (figure 9) with its 100k triangles has been processed by all methods (see below) in under a second on the same hardware.

Since there is a large body of previous work on salient edge extraction, we feel that it is necessary to conduct a thorough comparison. In the following we compare our technique to three ridge/valley schemes as well as to the only previous topology-based method. We are grateful to the respective authors for sharing their code or binaries with us.

5.1. Comparison to Ridge/Valley-Based Methods

Description of Methods The *Suggestive Contours* (SC) software package [DFR*] features a basic implementation of the ridge/valley definition [Thi96] applied to a curvature field computed by normal variation. The results are filtered by curvature thresholding.

The *crest lines* method of [YBS05] consists of two steps: computation of a smooth curvature field by local polynom fitting and tracing of curvature extrema. Curvature computation is steered by a parameter that defines the size of the neighborhood ring, i.e., the locality of the curvature. Furthermore, the software automatically smooths the input surface. The result can be filtered by so-called ridgeness, cyclideness, or sphericalness.

The *JavaView* software package features the method of [HPW05], where a discrete shape operator is used to obtain the principal curvatures. The method explicitly incorporates the possibility to smooth the extremalities. The user has to adjust the number of smoothing steps and the step size. Results are filtered by curvature thresholding.

Robustness to Noise Triangle meshes very often contain a certain amount of noise caused by different sources. Therefore, feature extraction methods should have a certain insensitivity to noise. This is especially important when the results shall be used as input for further computations. To assess the impact of noise on the extraction results of the different methods, we use a simple generic model where we know the undisturbed result and have full control over the amount of noise. We use a sphere that is scaled in z-direction as shown in figure 8. The structure we are looking for is the circumcircle in the xy-plane, which is a maximal line of κ_{max}. The objective is to reconstruct a closed circle for different amounts of noise, which is added to the model by displacing the vertices into normal direction. The amount of displacement is randomized and controlled by a parameter d, which scales the white noise $[0,1] \longrightarrow [0,d]$.

Figure 8 shows the results, where the rows denote different levels of noise and the columns represent the different methods. We did not include [YBS05] in this comparison as the software did not give meaningful results for this type of model – most likely an implementational issue and not a problem of the method itself.

Already the undisturbed model poses a problem for the straightforward ridge/valley implementation of the SC package as it can be seen in the middle of the first row: the

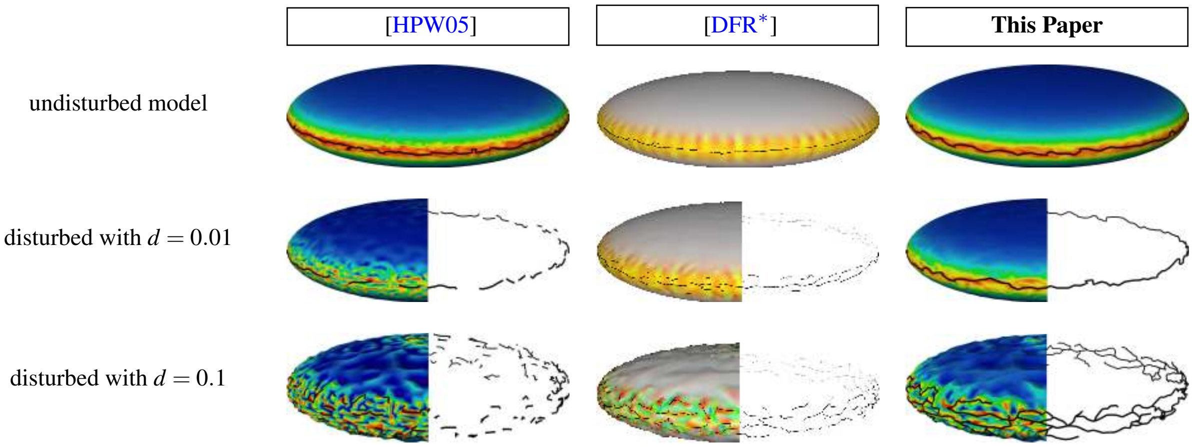


Figure 8: *Comparison of methods regarding their robustness to noise. All methods had to deal with the raw input, i.e., any smoothing of the surface or the curvature was forbidden. Curvatures have been computed in the default way for the respective method, i.e., [HPW05] uses its discrete shape operator, while our method and the SC package use normal variation. We used a clipping plane in the last two rows to expose the feature line. As it can be seen, our method is less noise-sensitive thanks to its global nature.*

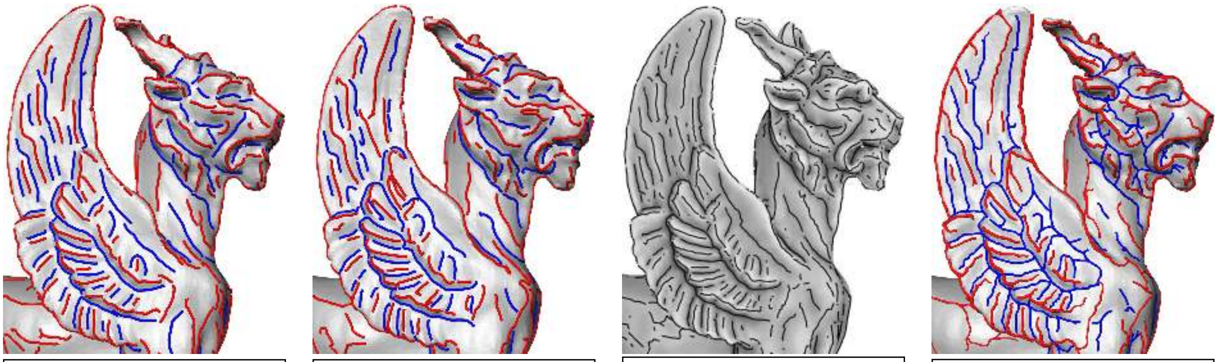
circle is a set of disconnected lines, whereas our method and [HPW05] are able to extract the feature as a single closed line. As it seems, a standard ridge/valley implementation can already be affected by the very small amount of noise introduced by the discretization of the analytic model. Also, the more robust computation of curvature derivatives in [HPW05] pays off. Note, however, that our curvature computation is comparable to the one used in SC (although the surfaces are colored differently, since the SC software always uses an advanced colormap showing $\kappa_{min}, \kappa_{max}$ while we just show κ_{max}).

In the second row we added a small amount of noise. Both ridge/valley methods are affected now, whereas our topological technique still captures a single closed line. We tried to push that a bit further in the last row by adding 10 times more noise. Still, our technique detects a closed structure (with a number of branches) and both ridge/valley methods are strongly affected. The reason why our topological method is much more robust against noise than ridge/valley methods lies in its global nature: local fluctuations are much less accounted for and removed early during topological simplification. In addition, the more we simplify the topological skeleton, the more it captures the overall shape of the model.

We are aware of the fact that smoothing the surface or the curvature would give nicer looking results for all methods. However, "nicer looking" is not a criterion in all applications and especially not, when the results are supposed to serve as input for further computations. Smoothing may very well dislocate or otherwise alter the features in an uncontrollable way. As the results of this comparison indicate, our method is preferable in applications where noise is an issue and smoothing is not desired.

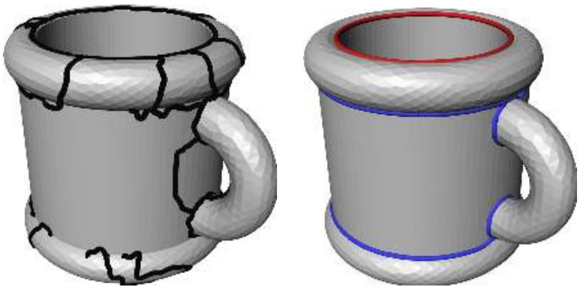
Parameter Dependence and Connectedness of Results In figure 9 we used the feline model to compare all three ridge/valley methods with our technique. This model features some challenging filigree structures on the wing. All methods are able to extract the most prominent edges. However, the basic ridge/valley implementation of the SC package shows the most noisy results with a lot of disconnected lines especially next to the head. Generally, our method generates longer lines. Note, how our lines form connected subnetworks on the wing and the head. We observed this difference to the ridge/valley-based methods in all our experiments and attribute it to the global nature of our approach.

Figure 9 also lists the parameters of every method that had to be set up in order to achieve the shown results. It is a major advantage of our method that the computation is completely parameter-free. This makes it easier to use, especially in batch jobs where parameter adjustments for individual meshes are prohibitive. Note, that the ridge/valley definition from [Thi96] itself does not impose computational parameters, since ridges and valley are well-defined mathematical constructs. The implementation in the SC package follows this road and does not add computational parameters. However, we find that the results of the SC package are less satisfying than those of [HPW05] and [YBS05]. Indeed, the latter two methods put more effort into an optimized computation of curvature derivatives – this is where the additional parameters come into play. The results in figure 9 clearly show that this pays off compared to the basic ridge/valley approach, but it also complicates implementation and application. Our method, on the other hand, yields qualitative similar results without derivatives and computational parameters – making it easier to use and implement.



[YBS05]	[HPW05]	[DFR*]	**This Paper**
Input			
curvature derivatives	curvature derivatives	curvature derivatives	**curvature**
Computational Parameters			
size of neighborhood ring	smooth extremalities num steps smooth extremalities step width	none	**none**
Visualization and Filtering Parameters			
ridgeness, cyclideness sphericalness	integral curvature local curvature	local curvature	**separatrix persistence**

Figure 9: *Comparison of methods regarding parameter dependence and the connectedness of the results using the feline model. Compared to ridge/valley-based methods, our technique yields qualitative similar results without derivatives and computational parameters.*



(a) Method of [SWPL08]. (b) Our method.

Figure 10: *Comparison of our method to the only previously existing topological method for salient edge extraction.*

5.2. Comparison to Topology-Based Methods

To the best of our knowledge, the only previously existing topological method for extracting salient edges is due to [SWPL08]. Their approach differs greatly from ours in three major aspects: first, they use the *Curvedness* measure $C = (\frac{1}{2}(\kappa_{min}^2 + \kappa_{max}^2))^{\frac{1}{2}}$ instead of the principal curvatures as the basis for their analysis, which makes it impossible to distinguish between concave and convex edges. Second, they propose an alternative to saddle persistence (cf. formula (1)), which considers integrals of C over separatrices and topological sectors, but with the disadvantage of loosing the monotone properties of persistence that are necessary for a consistent feature hierarchy. In fact, this makes it impossible to specify a threshold for feature strength and the user has to choose the number of cancellations instead. Also, remaining features may have a lower strength than previously removed ones. However, the most important difference to our approach regards the handling of separatrices: they have no means of quantifying the feature strength along a separatrix. Hence, a separatrix is considered either salient or non-salient in its entirety, which leads to a number of false positives and false negatives. Figure 10a shows this clearly. Our method classifies salient and non-salient parts of the skeleton correctly as shown in figure 10b.

6. Conclusions and Future Work

We presented the first topology-based approach for the extraction of view-independent salient edges, which is able to distinguish between salient and non-salient features correctly. The key behind this is the novel concept of *separatrix persistence*, which allows for the first time to handle the importance of a separatrix in a smooth manner. We conducted a thorough comparison with other state-of-the-art edge detection methods. Our extraction method has a global nature and applies derivative-free, purely combinatorial computations. Furthermore, it is parameter-free, easy to implement and fast.

While the differences between ridges/valleys and separatrices have been discussed in the literature [Ebe96,SWTH07, PS08] for general scalar fields, it is still an open question how these differences play out for curvature fields on surfaces: is it possible that one approach misses a salient edge found by the other? The answer is highly non-trivial due to the lack of a formal definition of salient edges not based on either approach. The examples in this paper suggest, however, that both approaches yield very similar results after thresholding. Also, we assessed robustness against artificial noise only. Another interesting question is how both approaches behave in the presence of non-artificial noise, e.g., coming from a laser scanner.

Our approach to quantifying topological separatrices is applicable to any two-dimensional scalar function. Many applications in other domains – such as image processing and scientific visualization – rely on a robust extraction of extremal lines, and could benefit from a customized variant of our technique.

References

[BP07] BREMER P.-T., PASCUCCI V.: A practical approach to two-dimensional scalar topology. In *Topology-based Methods in Visualization*. Springer, 2007, pp. 151 – 169.

[CCL03] CAZALS F., CHAZAL F., LEWINER T.: Molecular shape analysis based upon the Morse-Smale complex and the Connolly function. In *SCG '03* (2003), pp. 351–360.

[CP03] CAZALS F., POUGET M.: Estimating differential quantities using polynomial fitting of osculating jets. In *SGP '03* (2003), pp. 177 – 187.

[CSEH07] COHEN-STEINER D., EDELSBRUNNER H., HARER J.: Stability of persistence diagrams. *Discrete and Computational Geometry 37* (2007), 103–120.

[CSM03] COHEN-STEINER D., MORVAN J.-M.: Restricted Delaunay triangulations and normal cycle. In *Proc. ACM Symposium on Computational Geometry* (2003), pp. 237 – 246.

[DBG*06] DONG S., BREMER P.-T., GARLAND M., PASCUCCI V., HART J. C.: Spectral surface quadrangulation. *ACM Trans. Graph. (SIGGRAPH '06) 25*, 3 (2006), 1057 – 1066.

[DFR*] DECARLO D., FINKELSTEIN A., RUSINKIEWICZ S., SANTELLA A., BURNS M., KLAWE J.: Suggestive contours software. http://www.cs.princeton.edu/gfx/proj/sugcon/.

[Ebe96] EBERLY D.: *Ridges in Image and Data Analysis*. Kluwer Acadamic Publishers, Dordrecht, 1996.

[EHZ03] EDELSBRUNNER H., HARER J., ZOMORODIAN A.: Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry 30*, 1 (2003), 87 – 107.

[ELZ02] EDELSBRUNNER H., LETSCHER D., ZOMORODIAN A.: Topological persistence and simplification. *Discrete and Computational Geometry 28*, 4 (2002), 511 – 533.

[FG82] FIRBY P., GARDINER C.: *Surface Topology*. Ellis Horwood Ltd., 1982, ch. 7, pp. 115–135. Vector Fields on Surfaces.

[For98] FORMAN R.: Morse theory for cell-complexes. *Advances in Mathematics 134*, 1 (1998), 90–145.

[For02] FORMAN R.: A user's guide to discrete morse theory. Séminaire Lotharingien de Combinatoire, B48c, 2002.

[GH86] GUCKENHEIMER J., HOLMES P.: *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*, 2nd ed. Springer, 1986.

[GI04] GOLDFEATHER J., INTERRANTE V.: A novel cubic-order algorithm for approximating principal directions vectors. *ACM Transactions on Graphics 23*, 1 (2004), 45–63.

[GW01] GUSKOV I., WOOD Z.: Topological noise removal. In *Proc. Graphics Interface 2001* (2001), pp. 19–26.

[HH89] HELMAN J., HESSELINK L.: Representation and display of vector field topology in fluid flow data sets. *IEEE Computer 22*, 8 (August 1989), 27–36.

[Hos92] HOSAKA M.: *Modeling of curves and surfaces in CAD/CAM*. Springer, 1992.

[HPW05] HILDEBRANDT K., POLTHIER K., WARDETZKY M.: Smooth feature lines on surface meshes. In *SGP '05* (2005), pp. 85 – 90.

[IFP95] INTERRANTE V., FUCHS H., PIZER S.: Enhancing transparent skin surfaces with ridge and valley lines. In *IEEE Visualization '95* (1995), pp. 52 – 59.

[Lew02] LEWINER T.: *Constructing discrete Morse functions*. Master's thesis, Department of Mathematics, PUC-Rio, 2002.

[MDSB02] MEYER M., DESBRUN M., SCHRÖDER P., BARR A.: Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III*, Hege H.-C., Polthier K., (Eds.). Springer, 2002, pp. 35 – 57.

[MW99] MANGAN A. P., WHITAKER R. T.: Partitioning 3D surface meshes using watershed segmentation. *IEEE TVCG 5*, 4 (1999), 308 – 321.

[NWB*06] NATARAJAN V., WANG Y., BREMER P.-T., PASCUCCI V., HAMANN B.: Segmenting molecular surfaces. *Comput. Aided Geom. Des. 23*, 6 (2006), 495 – 509.

[OBS04] OHTAKE Y., BELYAEV A., SEIDEL H.-P.: Ridge-valley lines on meshes via implicit surface fitting. In *Proc. SIGGRAPH '04* (2004), pp. 609 – 612.

[PCM03] PASCUCCI V., COLE-MCLAUGHLIN K.: Parallel computation of the topology of level sets. *Algorithmica 38*, 2 (2003), 249–268.

[PS08] PEIKERT R., SADLO F.: Height Ridge Computation and Filtering for Visualization. In *Proc. Pacific Vis 2008* (2008), pp. 119 – 126.

[RM00] ROERDINK J., MEIJSTER A.: The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae 41*, 1-2 (2000), 187–228.

[SF04] STYLIANOU G., FARIN G.: Crest lines for surface segmentation and flattening. *IEEE TVCG 10*, 5 (2004), 536 – 544.

[Sty04] STYLIANOU G.: A feature based method for rigid registration of anatomical surfaces. In *Geometric Modeling for Scientific Visualization*. Springer, 2004, pp. 139 – 152.

[SWPL08] SAHNER J., WEBER B., PROHASKA S., LAMECKER H.: Extraction of feature lines on surface meshes based on discrete morse theory. In *Proc. EuroVis 2008* (2008), pp. 735–742.

[SWTH07] SAHNER J., WEINKAUF T., TEUBER N., HEGE H.-C.: Vortex and strain skeletons in eulerian and lagrangian frames. *IEEE TVCG 13*, 5 (2007), 980–990.

[Thi96] THIRION J.-P.: The extremal mesh and the understanding of 3D surfaces. *International Journal of Computer Vision 19*, 2 (1996), 115 – 128.

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