Topological Spines:
A Structure-Preserving Visual Representation of Scalar Fields

Carlos D. Correa, *Member, IEEE*, Peter Lindstrom, *Member, IEEE*, and Peer-Timo Bremer, *Member, IEEE*

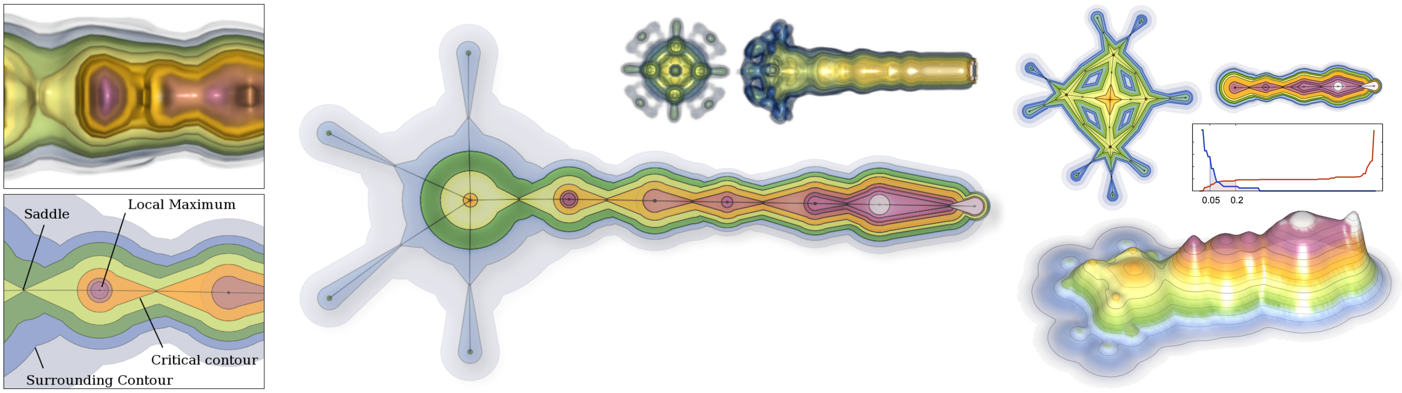


Fig. 1. Topological spine of fuel injection in a combustion chamber. Top left: Cutaway view of a volume rendering of the scalar field. Bottom left: a zoomed-in view showing the structural elements of a topological spine. Middle: A topological spine is a graphical representation of the topology that preserves the relative position of extrema and also conveys the nesting structure of the surrounding contours. On top we show two views of the scalar field. Right: Two persistence parameters—noise (blue) and variation (red)—can be explored to obtain a different topological spine, highlighting different structural aspects of the scalar field, such as symmetries and cycles. Extruding this structure in 3D allows us to create structure-preserving topological landscapes.

Abstract—We present *topological spines*—a new visual representation that preserves the topological and geometric structure of a scalar field. This representation encodes the spatial relationships of the extrema of a scalar field together with the local volume and nesting structure of the surrounding contours. Unlike other topological representations, such as contour trees, our approach preserves the local geometric structure of the scalar field, including structural cycles that are useful for exposing symmetries in the data. To obtain this representation, we describe a novel mechanism based on the extraction of *extremum graphs*—sparse subsets of the Morse-Smale complex that retain the important structural information without the clutter and occlusion problems that arise from visualizing the entire complex directly. Extremum graphs form a natural multiresolution structure that allows the user to suppress noise and enhance topological features via the specification of a persistence range. Applications of our approach include the visualization of 3D scalar fields without occlusion artifacts, and the exploratory analysis of high-dimensional functions.

Index Terms—Scalar field topology, topological spine, extremum graph, Morse-Smale complex.

1 Introduction

Topological analysis has become an important branch of scientific visualization. Not only does the topology of scalar fields encode important properties such as the number of connected components of an iso-surface or their critical points, but various topological graphs can be used as abstract representations of the underlying data set. Thus, the visualization community has witnessed a number of approaches that use topological structures as a visual summary of a complex data set. Examples of these include the contour tree, which abstracts the way level sets merge and split [6]; Reeb-graphs, which extend contour trees to more general manifolds, potentially creating loops [25]; and Morse-Smale (MS) complexes, which decompose the function in terms of regions of uniform gradient flow [9].

Representations like the contour tree and the Morse-Smale complex can be described as a graph, where nodes indicate critical points and arcs represent various notions of neighborhood in the domain based, for example, on the function range or steepest lines. As a structural summary, these graphs provide a decomposition of the domain that can be used, for example, for feature detection [15], transfer function design [29], data mining [12], and regression [13]. To be effective as a visual representation, however, topological graphs are typically used in conjunction with direct volume rendering or scatterplots, since the graphs alone are either too abstract or too complex to effectively convey the structure of the data set. As a result, these graphs are not well suited to support tasks that demand spatial inference.

In this paper, we introduce a new visual representation aimed at preserving both topological and structural (geometric) properties of a scalar field, called *topological spines*. Topological spines link together chains of critical points using canonical visual representations that preserve the topology and locality of extrema and the nesting structure of the surrounding contours. Locality in this context is defined as the neighborhood in the Morse complex, which is roughly equivalent to the structure of ridges/valleys connecting maxima/minima. Since topological spines preserve this important geometric information, they are better suited for spatial reasoning, support queries that demand knowledge about the spatial proximity of critical points, and provide global information such as the presence of symmetries.

The challenge to create such a visual representation is twofold: On

- *Carlos D. Correa, Peter Lindstrom and Peer-Timo Bremer are with the Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory. {correac,pl,bremer5}@llnl.gov.*

Prepared by LLNL under Contract DE-AC52-07NA27344.

Manuscript received 31 March 2011; accepted 1 August 2011; posted online 23 October 2011; mailed on 14 October 2011.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org.

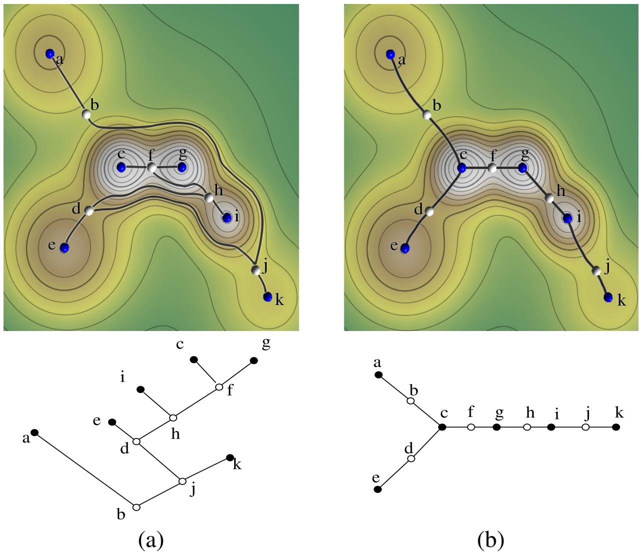


Fig. 2. (a) The contour tree (bottom) of a simple 2D terrain describes how contours merge and split to form individual components. However, embedding the arcs of the tree into the domain (top) can connect points arbitrarily far away (e.g., $b-j$). (b) An extremum graph connects critical points along steepest ascending (or descending) lines, which join adjacent extrema and therefore better preserve locality. Even an abstract representation of the extremum graph (bottom) retains some of the geometric structure of the 2D terrain.

one hand, even as the dimensionality and complexity of data sets rise, we aim for a representation that succinctly abstracts the way level sets are nested and critical points are related. However, as the dimensionality of the data set increases, less and less of this information can be preserved by a visualization, and global properties, such as density, symmetry, and periodicity of features, often disappear. On the other hand, representations that capture all these structural properties, e.g., the Morse-Smale complex, are difficult to visualize directly without severe occlusion problems, especially for higher-dimensional data sets. The challenge thereby lies in finding a representation that can balance the need to preserve topological and geometric relationships of critical points while avoiding excessive clutter and occlusion.

To address this problem we introduce *extremum graphs*, a simplified substructure of the Morse-Smale complex that encodes how neighboring extrema are connected via “ridge”- or “valley”-like saddle points. The result is an intuitive structure that retains the simplicity and readability of contour trees, as well as the geometric proximity of the Morse-Smale complex. As an example, consider the terrain in Fig. 2(a) and its corresponding contour tree. Arcs in the contour tree primarily indicate proximity in function space, and the only geometric information is that an arc describes neighboring contours of the same genus. Since contours can be arbitarily large and complex, embedding arcs in the domain (top) can lead to long unintuitive connections, e.g. $b-j$. In its abstract representation (bottom), however, contour trees lose virtually all geometric structure, and critical points in close proximity in the domain may appear far apart in the tree, e.g. $i-j$. A more natural way to connect these critical points is the extremum graph (see Fig. 2(b)), which represents the ridge-like connections between extrema (maxima in this case) as given by the Morse complex. Even in its abstract form the extremum graph preserves much of the local structure and overall shape of the original features. Additionally, extremum graphs can be defined hierarchically using two parameters: one that helps discard structures that may be considered as unimportant (often noise), and the other that helps disconnect parts into distinct ridge- or valley-like structures.

Topological spines are enhanced visual representations of extremum graphs that are augmented with geometric information and the nesting structure of the surrounding contours. Consider the scalar field in Fig. 1, which depicts fuel density in a combustion chamber after fuel is injected. Visualizing its structure accurately is important to understand how fuel mixes with air in the search of more efficient combustion mechanisms. In Fig. 1 we show a topological spine of the same data set, which preserves the Morse neighborhood of the extrema. In the inset, we see that the topological spine connects extrema via important structural saddles and depicts contours in an abstract form to retain the nesting structure of the level sets of the function. We provide a good depiction of this data set using planar graphs with minimal or no occlusion artifacts while exploring the two structural parameters. In Fig. 1(right), we show the resulting topological spine for a different persistence interval. The scalar field is now explained as two structures: one as the turbulent mix of air and fuel (note the radial symmetry of extrema), and another as the tubular structure of the fuel injection. At the bottom, we render the topological spine as a terrain or topological landscape [34], where each peak corresponds to an extremum, with height proportional to scalar value.

We show that topological spines are useful for abstracting the shape and structure of complex 3D scalar fields, such as the electron density and electrostatic potential of simple molecules and proteins. They also act as 2D maps that facilitate feature selection and tracking, which are challenging tasks in higher dimensions, and the resulting shapes offer insight about the hierarchical, fractal, or cyclical structures of multidimensional functions.

2 Related Work

Topological analysis has received significant attention in the graphics and visualization communities for both scalar and vector valued data. The initial research has been primarily focused on computing topological structures such as the topological skeleton for vector fields [18,30], contour trees [6,33], Reeb graphs [25,28,31], or Morse-Smale complexes [9, 14]. Subsequently, topological structures have been used to enhance existing visualization techniques [7,35] as well as for scientific analysis [5,20].

A more recent trend is to visualize topological structures directly in order to construct a more abstract view of various types of data. In particular, the ability of some structures to generalize beyond three dimensions provides an attractive alternative to standard dimension reduction techniques. A natural candidate is the contour tree, which can be computed in any dimension [6] yet can always be embedded in the plane. However, for practical data sets contour trees quickly become too large to be displayed [7], and a proper layout can take hours to complete. While hierarchical techniques as well as three-dimensional layouts such as the TopoOrrery [23] alleviate some of these issues, the resulting graphs remain too complex to be comprehensible for all but the simplest data sets. Reeb graphs and the Morse-Smale complex are even more detailed representations of a function’s topology, and thus any direct visualization suffers similar problems.

Instead, the focus has shifted to develop new visual metaphors to address the issue of visual complexity, or to selectively display carefully chosen subsets. Weber et al. [34] propose using topological landscapes to visualize contour trees. A topological landscape is a terrain whose contours have the same function values and nesting relation as those of a potentially high dimensional data set. Thus, one can exploit the human ability to quickly understand topographic information to convey structural information of a function. This concept has been extended by Harvey and Wang [17] to preserve the relative volume of sets of contours, and used by Oesterling et al. [22] to visualize high dimensional density distributions. However, as discussed in Section 3, contour trees by definition do not preserve spatial locality, making the landscapes difficult to interpret.

The Morse-Smale complex on the other hand preserves locality, but in its entirety is far too complex to provide a meaningful visualization. To address this problem, Gerber et al. [13] propose to use a dual representation in which each Morse crystal (cell of highest dimension) is represented by a single arc connecting its minimum with its maximum. The resulting graph is then embedded in two or three dimensions. While representing only crystals reduces the number of elements that are drawn, it does not address the inherent problem of in-

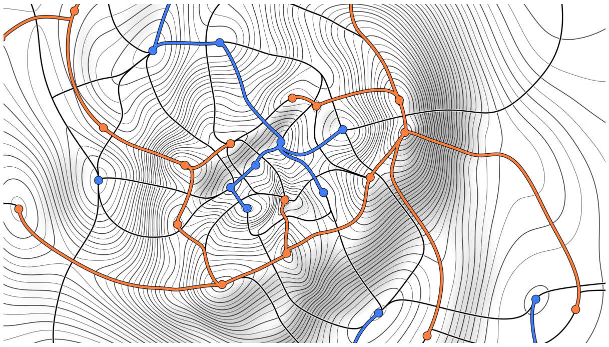


Fig. 3. Cancellation trees (wide blue and red lines for the minimum and maximum trees, respectively) and remaining arcs in the Morse-Smale complex (thin lines) for the terrain indicated by the contour lines.

creasing valence and the non-planarity of the graphs of higher dimensional functions. Furthermore, in many applications crystals remain a somewhat unintuitive representation. For example, a single mountain might be represented by many crystals, and spatial neighborhoods between mountains are not easily understood. Similarly, Beketayev et al. [3] use only the maxima and the arcs of the Morse-Smale complex connected to them to represent energies in chemical systems. Unlike Gerber et al. [13] they represent no geometric information and also have no control over valency and thus planarity.

Instead, we propose a simple hierarchical framework that allows the user to interactively explore structures with varying complexity. Additionally, we enhance the display by representing selected contours, their nesting relation, and their volume to provide a complete overview of the given function. As discussed in Section 3, we select arcs that represent important neighborhood relationships between extrema. This approach, while aiming to preserve different information, can be seen as a high dimensional extension to recent topology based techniques to extract feature lines from surfaces [26, 36].

3 Extremum Graphs

As mentioned above, the topological spine represents a carefully chosen subset of the topological structure of a given function. In this section we introduce the necessary theory to define *extremum graphs*, which describe the connectivity of the spine, and introduce an algorithm to compute them in arbitrary dimensions. Our goal is to find a balance between a *complete* representation, like the Morse-Smale complex, and an easily *comprehensible* representation, like the cancellation tree [4], which is guaranteed to be a planar graph. Properties that should be preserved are relative location of extrema as well as the neighborhood relation with respect to the gradient lines connecting them. First, we define the Morse-Smale complex, which forms a convenient superset of information from which to extract the extremum graphs. Subsequently, we describe an algorithm to compute *optimal* extremum graphs based on L_∞-optimal simplification of the Morse-Smale complex [10]. Finally, we discuss the approximations necessary to compute extremum graphs in higher dimensions.

3.1 Morse-Smale Complex

Let $\mathbf{M}$ be a smooth, n-dimensional manifold without boundary, and let $f : \mathbf{M} \to \mathbb{R}$ be a smooth function with gradient ∇f. A point $x \in \mathbf{M}$ is called *critical* if $\nabla f(x) = 0$ or *regular* otherwise. A function is called *Morse* if all critical points have pairwise different function value and their Hessian matrix of second derivatives is non-degenerate. Given a Morse function f the *index* of a critical point u is defined as the number of negative eigenvalues of its Hessian matrix. Minima have index 0, maxima index n, and there exist $n-2$ types of saddle points. At any regular point x the gradient (vector) is well-defined, and integrating it in both directions traces out an *integral line* $\gamma(s)$, $\frac{d}{ds}\gamma(s) = \nabla f(\gamma(s))$, which is said to start at a minimum and end at a maximum. The ascending/descending manifold of a critical point u is defined as all

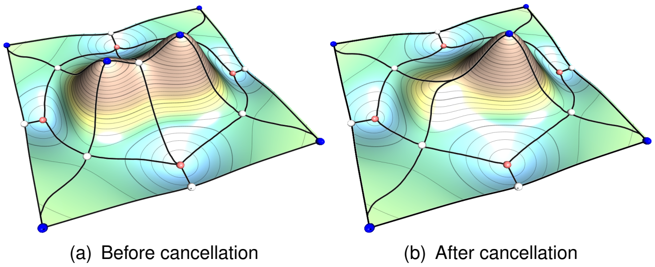


Fig. 4. Cancellation of a saddle-extremum pair. Note how the two arcs orthogonal to the cancellation are removed, while the other two remain in the simplified complex.

points whose integral lines start/end at u. The descending manifolds form a complex called the Morse complex, and the ascending manifolds define the Morse complex of $-f$. Given some additional nondegeneracy conditions, the intersections of ascending and descending manifolds create the Morse-Smale (MS) complex of f (see Fig. 3).

The MS complex consists of n-dimensional polygonal cells called *crystals* whose nodes are critical points and whose arcs are integral lines connecting neighboring critical points that differ in index by one. In practice, the complex often contains noise and/or features at different scales, and thus techniques have been developed to simplify it. Two critical points connected in the MS complex can be *canceled* by removing both nodes and reconnecting the remaining complex (see Fig. 4). Cancellations are typically ordered by *persistence*—the difference in function value between the two canceled nodes—which results in an L_∞-optimal topological simplification of f. While this operation is fairly straightforward in two dimensions (see Fig. 4), cancellations become significantly more involved in higher dimensions [15], and a detailed discussion is beyond the scope of this paper.

The MS complex completely describes the gradient flow of f, making it an ideal starting point for analysis. However, for visualization purposes the completeness is a significant drawback. Even without increasing the number of extrema, the number of arcs in the MS complex can increase exponentially with increasing dimensions. Intuitively, additional dimensions provide more opportunities for structures to be neighbors, which expresses itself in additional arcs. Furthermore, noise can create a large number of additional saddles, each of which creates a large number of additional arcs. Thus, displaying even simplified complexes of relatively simple, three-dimensional functions [15, 16] results in convoluted images that are difficult to interpret. However, to gain a high level global understanding of the topological structure, much of the information in the MS complex is unnecessary or even detrimental.

The structures of interest in a scalar function are almost always associated with extrema, and typically only with either maxima or minima. Thus, displaying only the portions of the MS complex connecting (one type of) extrema provides a first step towards reducing the visual complexity without losing important information [3]. However, the number of arcs can still be substantial. In particular, as shown in Fig. 6(a), there often exist multiple arcs that connect the same (set of) extrema. The two-dimensional example of Fig. 6(a) shows how noise in regions of minima creates a large number of arcs connected to maxima. Similar problems arise in all dimensions, where noise in the $(n-2)$-saddles creates additional arcs connected to maxima.

3.2 Cancellation Trees

When studying maxima our goal is to select arcs corresponding to neighborhoods relevant to maxima. One way to define a minimal set of neighbors are *cancellation trees* introduced by Bremer et al. [4]. Given an MS complex they perform a persistence based simplification, which successively merges neighboring critical points. The cancellation tree corresponding to a surviving maximum m is defined as all maxima and saddles merged with m, as well as the arcs between them (see Fig. 3). Intuitively, during the simplification each extremum

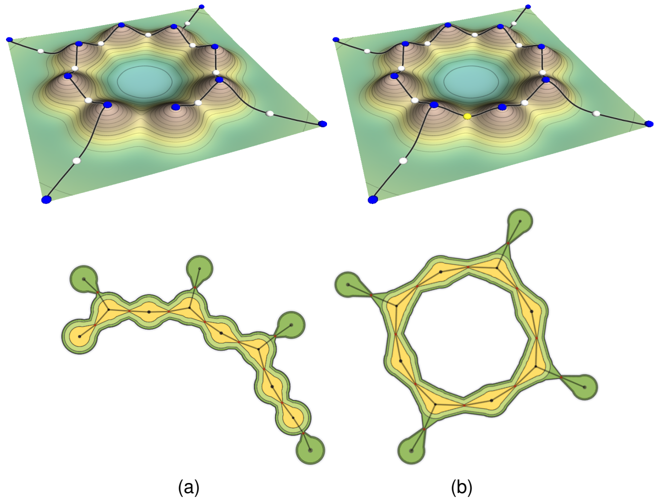


Fig. 5. (a) Cancellation tree of maxima for a cycle of hills (top) and its topological spine (bottom). Since a cycle cannot be canceled, the front most saddle is part of the minima cancellation tree, and the spine provides a misleading structure. (b) The maximum-graph of the same terrain. Since the front most saddle (yellow) is closer in function value to the maxima than to the center minimum, the maximum-graph closes the cycle and creates a more appropriate topological spine (bottom).

picks its most likely neighbor by the direction that an L_∞ optimal simplification would direct its gradient flow. As shown in Fig. 3, by simultaneously simplifying the entire MS complex most spurious arcs are removed, and the resulting cancellation trees represent ridge- and valley-like structures.

However, the construction of cancellation trees using only cancellations is too restrictive. Consider, for example, the terrain shown in Fig. 5. The maxima describing the rim will naturally merge until only a single representative remains (Fig. 5(a)). Since a maximum cannot be canceled with itself, the remaining saddle will be used to cancel the interior minimum, making it part of a minimum-tree. This breaks the symmetry of the data and to some extent misrepresents it. Intuitively, the saddle should be part of the rim, yet global topological constraints prevent the cancellation trees from creating such a structure. To address this problem we introduce *extremum graphs* that describe a set of unconstrained neighbors.

3.3 Extremum Graphs

Instead of canceling critical points we note that, at least in two dimensions, each saddle is either part of a minimum or a maximum cancellation tree. In general, saddles that appear more ridge-like belong to the maximum-tree, while saddles that are more valley-like belong to the minimum-tree. However, as shown in Fig. 5, in cancellation trees this property is superseded by constraints on the topological validity of cancellations. We remove these constraints by directly classifying each saddle as either connecting neighboring maxima (creating a ridge) or neighboring minima (creating a valley) via cancellation. Since we are interested in the global topological structure, we use an extended notion of persistence for classification, but other metrics such as a high dimensional version of separatrix persistence [36] could also be applied.

Computation. Starting from a two-dimensional MS complex the L_∞-optimal extremum graphs of minima and maxima can be constructed by a slight variation of the standard persistence simplification. As usual, all saddles are entered into a queue sorted on persistence. However, unlike the traditional algorithm we compute a "persistence" not just for valid cancellations but also for potential strangulations (saddles connected to the same extremum twice). In each iteration we choose the saddle with lowest persistence and classify it depending on whether its partner is a maximum or a minimum. If the pair of critical points correspond to a valid cancellation they are simplified and removed from the graph. Otherwise, only the saddle is removed. The extremum graph of maxima/minima is then defined as the collection of all maxima/minima plus the saddles classified as ridges/valleys and the arcs connecting them. Note that this strategy naturally introduces cycles, since, for example, the persistence of the last rim-saddle of Fig. 5 is lower towards its maxima than towards its minima. The same algorithm applies in dimensions beyond two, except that there exist additional classifications. More specifically, an n-dimensional function defines n types of graphs: one for each successive index pairing. However, as discussed above, we are typically only interested in the $[0,1]$ (minima) graphs and $[n-1,n]$ (maxima) graphs.

Since the extremum graphs are constructed using a simplification like procedure, it is natural to use the resulting persistences to remove noise and/or small scale features. As discussed in more detail below, we store persistences with each critical point and at run-time display only critical points above a user defined threshold. Deciding which arcs to display is slightly more involved, and will be described as part of the high dimensional framework.

3.4 High-Dimensional Approximations

The biggest drawback of the algorithm discussed above is the fact that MS complexes are expensive to compute and handle. While a general n-dimensional algorithm has been proposed [14] it has never been demonstrated beyond three dimensions. Furthermore, it is not obvious how to extend cancellations to higher dimensions, and even if possible the resources required to compute and store the complex for large data sets would be impractical. Here we describe how to create an approximate MS complex in arbitrary dimensions, and show how this complex naturally creates approximate extremum graphs. Furthermore, we introduce an algorithm to encode extremum graphs in a multi-resolution fashion, which helps recover some of the useful properties of the optimal extremum graphs.

Approximate MS Complex. As shown, for example, in [8, 13], it is easy to segment data of arbitrary dimension into ascending and descending manifolds corresponding to the extrema. Furthermore, for each neighboring pair of manifolds the highest/lowest shared point between descending/ascending manifolds must be an $(n-1)/1$-saddle in the full MS complex. Since, especially in high dimensions, most other elements of an MS complex are difficult to extract we use this set of extrema, saddles, and connecting arcs as an approximation of the full MS complex. Clearly, the resulting approximate MS complex is only a subset of the true MS complex (see Fig. 6(b)) which lacks all intermediate saddles of index 2 to $(n-2)$ as well as the 1- and $(n-1)$-saddles that either create small loops (are connected to the same extremum twice) or that duplicate existing connections between a pair of extrema. Note that, two-dimensional functions are a special case since they contain only one type of saddle. Therefore, some saddles of the true MS complex shown in Fig. 6(a) are found to split both maxima and minima, shown as valence four nodes in Fig. 6(b). In dimensions higher than two this would not occur, and the high-dimensional approximations of the MS complex consists of two independent graphs: one containing minima and 1-saddles, and one containing maxima and $(n-1)$-saddles. In particular, one can view this approximation of the MS complex as a *directed gradient flow graph*, with arcs from saddles towards extrema.

As shown above, this directed *gradient flow graph* is a partial approximation of the exact MS complex and thus has missing elements. For example, we cannot detect small loops or multiple connections. However, given the presence of noise, it is unlikely that loops involving a single extremum are significant, and multiple connections primarily represent information about critical points unrelated to the extrema. For example, in Fig. 6(a) the multiple connections between maxima are due to noise around the minima. Since we aim to develop a high level visualization rather than a detailed analysis, the missing information is mostly non-essential. Missing elements have the effect of classifying saddles as either part of the maxima or minima graph: each saddle of index k is classified as belonging to the graph in which it has a lower persistence (either the $[k-1,k]$ or the $[k,k+1]$ graph).

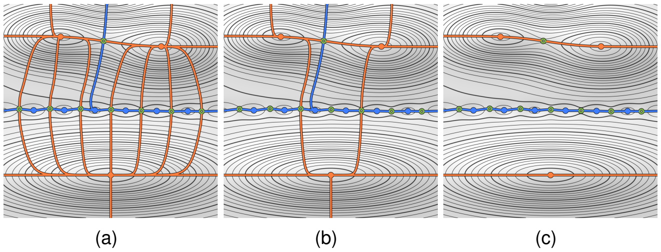


Fig. 6. (a) The Morse-Smale complex of two mountain ranges separated by a bumpy valley. The noise around minima creates additional saddle-maxima arcs. (b) The approximate Morse-Smale complex as computed by the high dimensional algorithm. Some saddles are detected separating both minima and maxima (valence four), while others only separate one type of extremum (valence two). (c) The extremum graphs of the Morse-Smale complex of (b), computed using a secondary persistence threshold, correctly detect the ridges and valleys.

Since the directed gradient flow graph only consists of components from the $[n-1,n]$ and $[0,1]$ graphs, each saddle has only one possible classification. As a result, all saddles of this approximation connected to maxima/minima are trivially added to the $[n-1,n]/[0,1]$ graphs. This behavior may connect extrema arbitrarily far away and with high persistences, as shown in Fig. 6(b), which would likely cancel in the full MS complex with $(n-2)$-saddles. To address this issue, we introduce an additional criterion to determine when a saddle should *not* be part of a graph and use it to approximate the extremum graph.

Approximate Extremum Graph. We construct the approximate extremum graph as a subset of the directed *gradient flow graph* that respects two thresholds based on persistence. One threshold, p_{lo}, controls *noise*, and all extrema with persistence below the noise threshold are simplified. A second threshold, p_{hi}, governs *variation*, and determines the extent to which a saddle is ridge-like or valley-like. Saddles with persistence above the variation threshold are removed. These two thresholds are seen in persistence plots throughout the paper as the lower and upper ends of a blue and red area plot, respectively.

The difference between simplification and removal is subtle. When we simplify a saddle, its incident arcs may still have to be displayed. Consider the volcano shown in Fig. 5(b). If after simplification the rim is represented by a single maximum, we wish to retain the arcs of the simplified saddles to mark the ridge. On the other hand, if we cut the cycle, as shown in Fig. 5(a), then we remove both the saddle in yellow and its incident arcs.

To achieve this goal, we compute two measures for each critical point: its persistence p and its *saturated persistence* $p^\star \geq p$. The persistence is computed during construction, and is at run-time compared against the noise threshold—for extrema—and also against the variation threshold—for saddles. Saturated persistence is slightly more involved. Conceptually, the saturated persistence of an arc a is the persistence of the last (highest-persistence) extremum u that needed a to be connected to the extremum graph. In practice, we compute saturated persistences by keeping track of the set of arcs that connect as yet unsimplified extrema. Each simplification reverses the arcs between the two canceled critical points, and thus creates new paths between the remaining critical points. At any time during the simplification, a saddle can find its current neighboring, unsimplified extrema by following the paths through its two outgoing directed arcs. When canceling a saddle s with persistence p, it is clear that all saddles along the arcs connecting s with its neighboring extrema must have a saturated persistence of at least p (since they are used by s for lower noise thresholds). We compare the saturated persistence with the noise threshold to determine whether a saddle must display its arcs. Because the two arcs incident on a saddle have the same saturated persistence $p^\star$, we store $p^\star$ with the saddle. The algorithm to compute both measures is shown in Algorithm 1. It takes as input the directed gradient flow graph and outputs a weighted extremum graph with persistences and saturated persistences for each critical point.

```
input  : Scalar field $f$ and gradient flow graph $G = (N,E)$
output: Persistence $P$ and saturated persistence $P^\star$ for each node
$P(N) \leftarrow P^\star(N) \leftarrow \infty$                                  Initialize persistence
$Q \leftarrow \{\langle s,0\rangle : s \in \textsc{Saddles}(N)\}$   Insert all saddles into priority queue
while $Q \neq \varnothing$ do
    $\langle s,p\rangle \leftarrow \textsc{Pop}(Q)$                          Pop lowest-persistence saddle
    $\langle u,v\rangle \leftarrow \textsc{Neighbors}(G,s)$                              Adjacent extrema
    $U \leftarrow \textsc{Path}(G,s,u)$
    $V \leftarrow \textsc{Path}(G,s,v)$                 Follow paths from $s$ through $u$ and $v$...
    $\hat{u} \leftarrow \textsc{Tail}(U)$                            ... to terminal extrema $\hat{u}$ and $\hat{v}$
    $\hat{v} \leftarrow \textsc{Tail}(V)$
    $p \leftarrow \min\{|f(\hat{u}) - f(s)|, |f(\hat{v}) - f(s)|\}$   Lazy-update persistence
    if $Q \neq \varnothing$ and $p > \min Q$ then          No longer lowest persistence?
        $Q \leftarrow Q \cup \langle s,p\rangle$                                   Reinsert into queue
    else
        $P(s) \leftarrow p$                                     Assign persistence to saddle
        $P^\star(\textsc{Saddles}(U \cup V)) \leftarrow p$         Update saturated persistence
        if $\hat{u} \neq \hat{v}$ then                               Do $U$ and $V$ not form a cycle?
            if $|f(\hat{u}) - f(s)| < |f(\hat{v}) - f(s)|$ then
                $P(\hat{u}) \leftarrow p$                                          Cancel $s$ with $\hat{u}$
                $\textsc{Reverse}(G,U)$                          Redirect flow from $\hat{u}$ to $\hat{v}$
            else
                $P(\hat{v}) \leftarrow p$                                          Cancel $s$ with $\hat{v}$
                $\textsc{Reverse}(G,V)$                          Redirect flow from $\hat{v}$ to $\hat{u}$
```

Algorithm 1. Approximate extremum graph computation. The function $\textsc{Path}(G,s,u)$ returns the sequence of critical points $(s,u,\ldots,\hat{u})$ on the maximal path in G starting with the arc su. $\textsc{Reverse}(G,P)$ modifies G by reversing the direction of all arcs on a path P.

Once we have determined these values, the extremum graph $G_R = \{N_R, E_R\}$ corresponding to a threshold range $R = [p_{lo}, p_{hi}]$ is formed as a subset of the gradient flow graph $G = (N,E)$:

$$\text{extrema } u \in N \qquad u \in N_R \iff R \cap [0, p(u)] \neq \varnothing \qquad (1)$$

$$\text{saddles } s \in N \qquad s \in N_R \iff R \cap [p(s), p(s)] \neq \varnothing \qquad (2)$$

$$\text{arcs } su \in E \qquad su \in E_R \iff R \cap [p(s), p^\star(s)] \neq \varnothing \qquad (3)$$

where p is the persistence associated with an extremum u or saddle s, and where $p^\star$ is the saturated persistence associated with the two arcs of a saddle s.

The structure described above provides a multi-resolution encoding of extremum graphs that allows a user to explore different resolutions and feature sizes. Raising the noise threshold removes low persistence extrema, while lowering the variation threshold prevents merging across high-persistence saddles. Note that an upper threshold may be useful even if the complete MS complex is known. There likely exist saddles which have high persistence in both their candidate graphs, in which case not connecting the corresponding critical points may be a more intuitive representation of the global topology.

4 Topological Spines

The extremum graph alone is a graphical representation that retains the structure of the scalar field, but does not convey information about the distribution and density of the function values, the size of features, or how contours merge. Thus, we desire a representation that can succinctly describe how the function values change in the domain. Such a representation should allow users to identify features in terms of size, density, or function value. To this end, we define the topological spine: a visual representation of the extremum graph augmented with geometric and contour nesting information. This representation consists of a collection of *canonical* visual elements, each of which encodes information about an extremum-saddle-extremum path. Since a scalar function defines both a maximum and a minimum graph, one can derive two topological spines for each function. For brevity we describe our method for the maximum-graph—the construction for the minimum-graph is analogous.

4.1 Canonical Topological Links

The basic element of a topological spine is a *canonical* topological link. We call it canonical, since it abstracts the nesting behavior and enclosed volume of (portions of) contours around an extremum-saddle-extremum triplet. As shown on top of Fig. 7, for each triplet we draw the contour through the saddle, called the *critical contour*, as a figure-eight shape (yellow) created by the geometric convolution (Minkowski sum) of a (infinite) set of circles centered on the extremum-saddle arc, with the radius of each lobe proportional to the enclosed volume. Optionally, we also draw additional pesudo-contours above the saddle contour (beige), indicating regions around the extrema, and below (green) indicating contours surrounding the triplet. As with the critical contour the radii of additional contours are determined by the corresponding volume, as defined in Section 4.2. The link to the volume ensures that contours at lower iso-values are guaranteed to have a larger radius than higher contours and thus preserve the correct nesting behavior. Neighboring triplets share extrema and by construction their canonical links agree at extrema. Finally, drawing higher contours on top of lower contours naturally creates an intuitive nesting structure, as shown in Fig. 1.

4.2 Link Radius

As discussed above, canonical links are created as a convolution of circles centered on the extremum-saddle arc. To define their radii we first assign a function value to each point along the arc by linear interpolation between the function value of the maximum and the saddle. Subsequently, given a maximum-saddle arc us and an isovalue c the radius at a point $p \in us$ is computed as the volume enclosed between $f^{-1}(c)$ and $f^{-1}(f(p))$ restricted to the descending manifold of u:

$$r_{us}(c,p) = |\{x \in D(u) : \min(c, f(p)) < f(x) < \max(c, f(p))\}|.$$

Fig. 7 shows several examples for different contours and locations. In the second row the critical contour at $f(s)$ is shown at the maximum with a radius equal to the area enclosed by the corresponding contour of the function. The third row shows the critical contour at the half way point between u and s with a radius equal to area between the contours at $f(s)$ and $\frac{f(u)+f(s)}{2}$. Since $c = f(s)$ for the critical contour, the radius $r(f(s),s) = 0$, resulting in the figure-eight configuration typically associated with a saddle. The fourth and fifth rows show a contour with $c < f(s)$ at the maximum and the saddle, respectively.

4.3 Properties

Since the primary focus of topological spines is to provide insight into the topology and geometry of functions difficult or impossible to visualize directly (e.g. high dimensional functions) it is important to understand what properties of the source function are preserved and to what extent. In particular, one can identify three properties of interest:

Relative Location of Extrema. By construction, extrema graphs preserve the connectivity of high-persistence arcs in the Morse-Smale complex. Given an extremum, these can be seen as the most important neighboring extrema or, alternatively, those extrema connected by important ridges/valleys. While these arcs do not necessarily correspond to purely geometric distances/neighborhoods among extrema, they describe the "functional" neighborhoods defined by the gradient flow that are typically the structures of primary interest.

Volume Preservation. As described above, we augment all links with the information about the enclosed volume, and thus the relative thickness of the pseudo-contours corresponds to the relative volume enclosed by the corresponding contours in the domain.

Contour Nesting. By construction, we guarantee that in a maximum-graph, for example, pseudo-contours of lower isovalue always have a larger radius than those corresponding to higher isovalues. This in turn guarantees that the nesting of contours shown around the extrema and saddles is preserved. However, since extremum graphs use only a subset of the arcs of the Morse-Smale complex, the genus of these contours may not be preserved, as shown in Fig. 5. Furthermore, the nesting can only be preserved for planar graphs laid out without self-intersections. However, as we show in our results, depending on the resolution level of the structural parameters, we usually find a suitable planar representation of a scalar field.

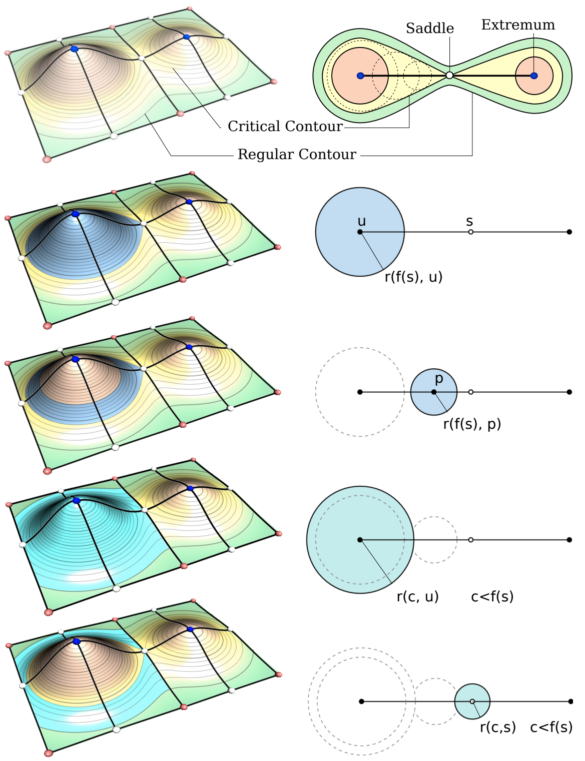


Fig. 7. Each region in the domain (left) gets encoded as a corresponding circle on the right. Rows 2 and 3 show the encoding of the critical contour $f(s)$ at the extremum u and a point p between the extremum and the saddle, where the radius of each circle is determined by the volume of the region formed by the descending manifold of u and the contours at $f(s)$ and $f(u)$. Rows 3 and 4 show the encoding of a larger level set with isovalue $c < f(s)$ that encloses the ones above.

4.4 Rendering

We draw the geometric structures computed using the mechanism described above for a series of isovalues. By default, we select all isovalues corresponding to the critical points of the scalar field. To avoid clutter, however, we enforce a minimum difference in isovalue that can be set by the user. We use two different color encodings in our visualization, selected by the user: by scalar value, as shown in Fig. 1, or by topological segment, as shown in Fig. 9(d). Colors and opacities are changed in real-time via a transfer function editor. The locations of the vertices of the topological spine are computed using a force-directed layout algorithm, which models the topological spine as a mass-spring model. To ensure that distances are somewhat preserved, the rest lengths of the springs are proportional to the Euclidean distance between two critical points. Alternative layouts for graphs can be used [32].

4.5 Multi-resolution Topological Spines

As described in Section 3, we express a family of extremum graphs in terms of two parameters: a lower persistence threshold p_{lo}, which is used to filter out noise, and an upper threshold p_{hi} (or variation), which is used to separate distinct features that if joined would not be representative of ridge-like structures. These two persistence thresholds allow exploration of the multi-resolution representation of topological spines. To find suitable values for these thresholds, we look at stability regions of the *extended persistence plot*, as shown in Fig. 8(b). This plot consists of two curves: A persistence plot (blue), which shows the number of extrema remaining in the extremum graph after simplification using a given noise threshold p_{lo}. Plateaus in this plot indicate regions where the structure is stable. Here, we compute the topological spine of the electron density of the cyclohexane molecule, formed by a cycle of twelve hydrogen and six carbon atoms (Fig. 8(b)). We see two stable regions: one starting at $p_{lo} = 0.015$, at which noisy features are removed, and another starting at $p_{lo} = 0.1$, at which the hydrogen atoms are removed. The second curve is a variation plot (red), which shows the number of arcs that remain in the graph after pruning those with persistence larger than the upper threshold p_{hi}. Similarly, plateaus indicate regions where the ridge-like features are stable. The stable region starting at $p_{hi} = 0.32$ corresponds to the structure that connects the carbon atoms. In the stable region between 0.1 and 0.28, the molecule is broken into six CH_2 components. Fig. 8 shows the resulting topological spines for two lower and three upper bounds.

5 Results

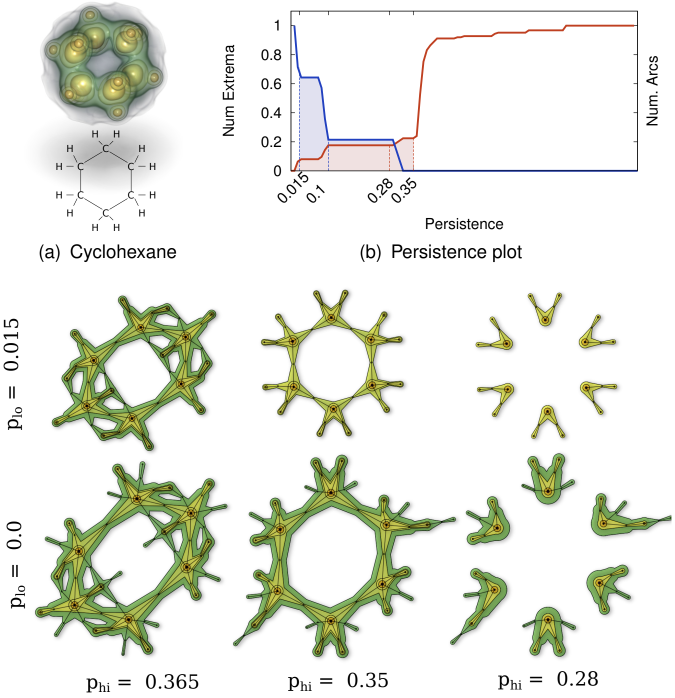
We have applied our technique to the analysis and visualization of scalar fields and the exploration of high-dimensional functions.

Geometric Structure from Scalar Fields. Our representation allows us to show complex geometric structures as 2D graphs. We have explored the use of our representation for a series of scalar fields. Fig. 1 shows the structure of fuel injection in a combustion chamber. We present two representations . The main representation in the middle succinctly summarizes the structure as two distinct parts: one consisting of the fuel injection itself, where the density of fuel is high (compared to air), that can be represented as a rod, and the other consisting of a radially symmetric expansion, where fuel mixes with air and produces turbulence. After exploring the two persistence parameters, we obtain a different topological spine (right), which reveals a more complex structure for the turbulent region, formed as two concentric rings of medium to low fuel density.

Fig. 9(a) shows volume renderings of the negative high potential of a protein. Color encodes the descending manifold of each extremum (top) or function value (bottom). Figs. 9(c–d) show the topological spines with color encoding function value and topological segmentation, respectively. Our approach extracts the symmetries of the scalar field, which are usually not preserved in the contour tree or the cancellation tree, since we are able to extract a cycle of extrema corresponding to the ring of atoms in the molecule. Fig. 9(b) shows a structure-preserving topological landscape, as described in detail below.

Fig. 10(right) shows the topological spine for a vortex simulation. The color represents vorticity for the 3D rendering on the top left, and segmentation on the bottom left and on the right. The fact that the resulting representation does not show a particular symmetry reveals the turbulent nature of the simulation. Nonetheless, since the spine is a 2D representation, it becomes easier to track and connect adjacent features in this structure than in the 3D space directly or using the MS complex, which would result in an overwhelming number of intersecting edges. Using our representation, we can select groups of adjacent extrema and tag them as features of interest. The blue circled extrema in the spine are highlighted in the 3D rendering, for instance.

Fig. 12(a) shows a scalar field representing the carbon molecule C_{60} (Buckminster Fullerene), also known as the Bucky ball. Fig. 12(b) shows the persistence plot and a planar graph describing the same structure. If we preserve the distances in the topological spine (Fig. 12(c)), the resulting structure contains intersections and may be difficult to read. Nonetheless, we can still see the projected structure. Using the entire MS complex approximation (i.e., setting $p_{hi} = \infty$) results in a highly non-planar graph (Fig. 12(b)), and any layout of such a graph would prevent us from recognizing the Bucky ball structure. If we do not enforce the preservation of distances for all edges, we can "open" the Buckyball and obtain a representation (Fig. 12(d)) more similar to the canonical planar graph representation (Fig. 12(b)).



(a) Cyclohexane (b) Persistence plot

(c) Matrix of extremum graphs

Fig. 8. Multi-resolution exploration of the density field of a cyclohexane molecule. (a) Electron density and schematic representation of the molecule. (b) Persistence plot, highlighting the region of persistence (blue) and saturated persistence (red) of interest. (c) Grid of multi-resolution representations for two noise thresholds (top: $p_{lo} = 0.015$; bottom: $p_{lo} = 0$) and three variation values. The most informative spine is at $p_{lo} = 0.015$, which gets rid of noise, and $p_{hi} = 0.35$, which connects the carbon bonds, and is found as the first plateau in the red curve around $p = 0.35$.

Our approach has applications in crystallographic topology. The topology of electron density is known to describe the structure of a molecule and can be explained by the quantum structure [2, 11]. Pascucci et al. [24] show that solutions based on contour trees fail to make a distinction between chains and rings or other complex structures, and seldom produce a structure that resembles the canonical (abstract) chemical notation. Leherte et al. build a critical point graph based on domain knowledge and geometric constraints to extract the primary 3D structure of proteins [21]. Johnson et al. suggest that the connections between critical points can indeed be extracted from the Morse complex of electron density [19]. However, as we point out, this results in a number of additional bonds that are not representative of a protein structure. Topological spines have the additional benefits of showing contour nesting and representing a family of structures, which together can lead to a better understanding of a complex molecule. Fig. 8 shows the structure extracted from the cyclohexane molecule. Similarly, Fig. 11 shows the topological spine and topological landscape for isopentane, resembling the schematic of the molecule.

High-dimensional Data Exploration. Topological spines easily generalize to higher-dimensional data sets. One of its applications is in optimization, where we are often interested in finding the structure of the local extrema of non-convex functions. The lack of structure-preserving visualization algorithms has prevented scientists and engineers from exploring these functions visually in higher dimensions.

Fig. 13(a) shows the topological spine for the 2D, 3D, 4D and 5D Ackley's path function. This function was designed by Ackley to test hill climbing strategies and other algorithms [1]. The function has a well-defined structure: it has a deep global minimum at $x = (0,0)$ in the middle of progressively shallower local minima. In 2D, we find eight local minima, which together with the global minimum form a 3×3 grid. In higher dimensions the local minima can be found as the cartesian product of three minima. Therefore, we should expect 3^d extrema in total and a global minimum connected to $2d$ extrema, one in the positive and negative direction along every dimension. In the 2D case, the planarity of the graph allows us to afford the depiction of the cyclical structure of the extrema. However, for higher

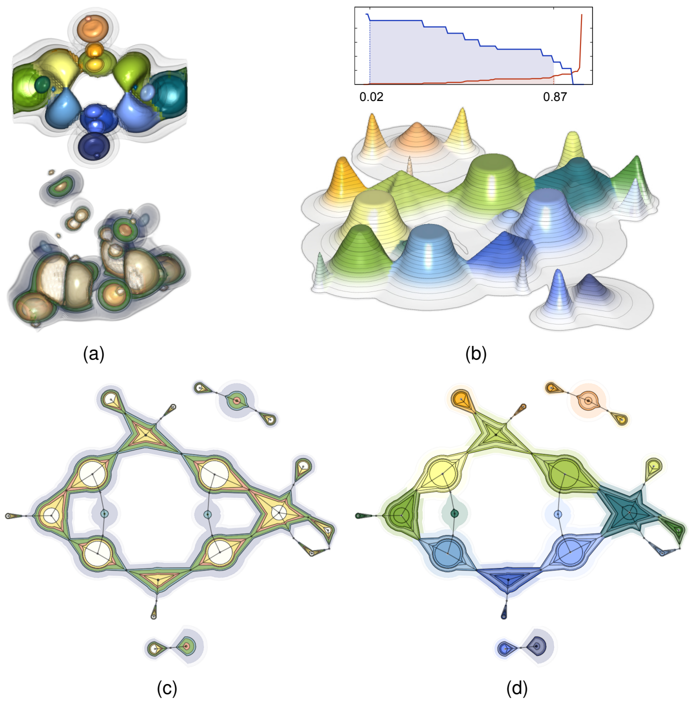


Fig. 9. Neghip data set (a) volume rendering, where color denotes each descending manifold (top) or scalar value (bottom). (b) Persistence plot and topological spine as a landscape (c–d) topological spines depicting scalar value (c) or topological segments (d). Our topological spine preserves the symmetry and cyclic structure of the molecule.

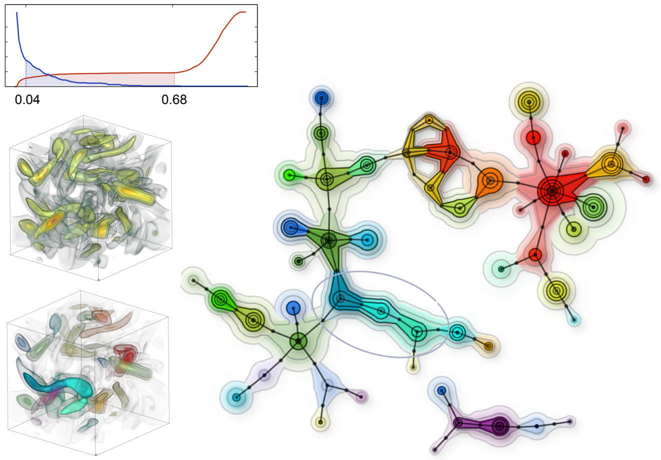


Fig. 10. Vorticity. Color coding each descending manifold allows us to identify and select features with ease. The planarity of the spine eliminates occlusion problems and facilitates feature selection. The circle contours in the topological spine allows us to select a features, highlighted in the 3D rendering in the bottom left corner.

dimensions it is easier to understand the function when we prune away high-persistence arcs that form cycles, since the resulting tree is now a 2D planar graph. This produces a clear view of the structure of the data set and the relative volume of the descending manifold associated with each extremum.

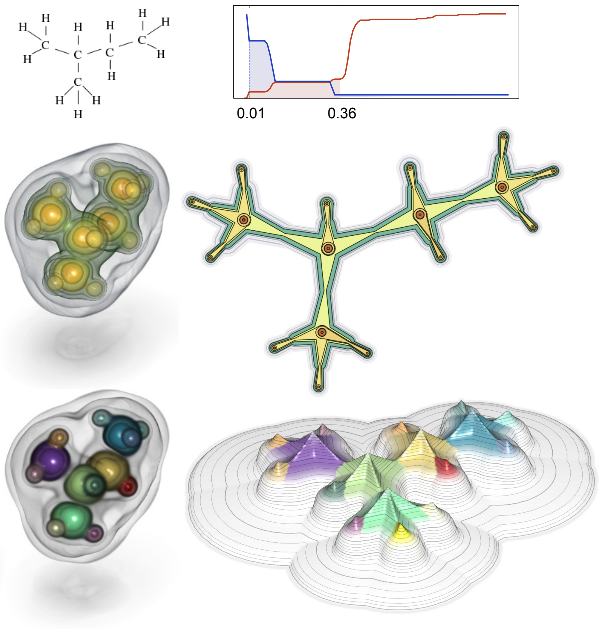


Fig. 11. Electron density of isopentane. Top: schematic and persistence plot. Middle: 3D rendering and corresponding topological spine, which accurately depicts the molecular structure. Bottom: 3D rendering with color encoding each descending manifold, and structure-preserving landscape, obtained by vertically extruding the spine.

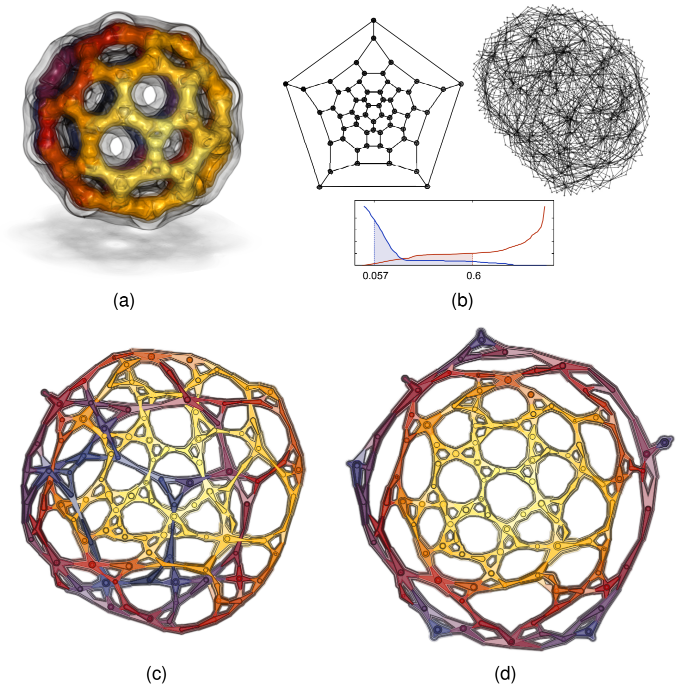
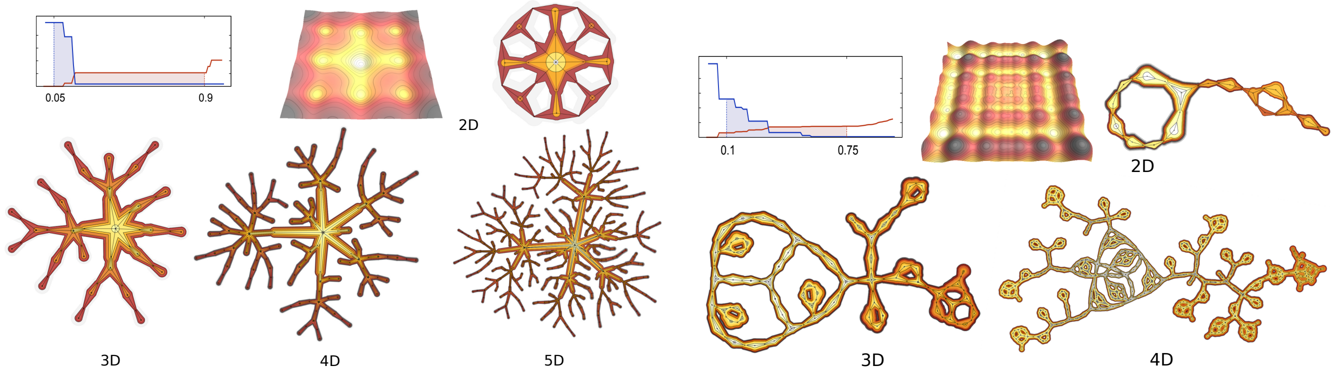


Fig. 12. (a) Bucky ball with color encoding descending manifolds. (b) Planar representation, 2D MS complex layout, and persistence plot. (c) A topological spine that preserves distances appears as a projection. Unlike in the 3D rendering, we now clearly see both the global structure and the local nesting of each bond. (d) By constraining the positions of nodes in one cycle, we can obtain a planar spine representation.

Fig. 13(b) shows Schwefel's multimodal function [27], whose local extrema are geometrically far apart in the domain. Similar to Ackley's path function, Schwefel's is designed for testing optimization algorithms. In 2D, this data set is formed by two cycles of extrema connected by a spiraling valley of minima. We show topological spines for the 3D and 4D functions, which highlight the fractal nature of this data set. The graphical shape formed by the arcs and cycles corresponds to geometric structures in the data that helps us predict what the function does in higher dimensions.



(a) Ackley

(b) Schwefel

Fig. 13. Topological spines of high dimensional functions. (a) Ackley's path function in 2–5 dimensions, which reveals structural properties, such as the $2d$-fold radial symmetry and the exponential increase in complexity (the number of extrema is 3^d). (b) Schwefel's function in 2D–4D. Note that the fractal structure and the cardinality of the different geometric structures are preserved.

Structure-preserving Topological Landscapes. Topological landscapes were introduced by Weber et al. [34] as intuitive visualizations of the contour tree of scalar fields. These landscapes leverage our ability to read and interpret elevation maps to understand the shape of a function. This metaphor has also been extended to higher-dimensional data sets [17, 22]. Since contour trees do not inherently encode spatial information, as shown in Section 3, proximity relationships in topological landscapes are usually arbitrary.

We here extend topological landscapes to preserve spatial locality. Our topological spine representation can be thought of as an approximate 2D contour map of the underlying scalar field. By extruding these contours in the third dimension according to their associated function values, we readily obtain a structure-preserving topological landscape.

One example is shown in Fig. 9(b) for the neghip data set. The shape and height of each hill here makes the density and persistence of each extremum more salient. Fig. 11(b) reveals the hierarchical structure of the density field of the isopentane molecule. The difference in scalar value of the H and C atoms (smaller and larger hills, respectively), is now more evident than in the 2D topological spine.

6 Limitations

Topological spines, and the underlying extremum graphs, offer a mechanism for displaying the structure and topology of a scalar field. Due to its construction, we consider the maximum and minimum graphs as independent. Therefore, the interplay between maxima and minima may not be evident from looking at the two corresponding topological spines. Topological spines and their landscape counterparts are most effective when the extremum graph is reduced to a planar graph. We have seen that this is the case for many complex structures such as the Bucky ball. In fact, many of the MS complex arcs that make these structures non-planar connect extrema via non-ridge and non-valley like saddles, and seldom help understand the structure. Nonetheless, there may be cases where the visualizations of these connections is important. Our algorithm allows the user to explore the two persistence thresholds, noise and variation, at runtime, and enables the exploration of structures from the global MS complex to more meaningful local structures. Topological spines rely on setting two persistence thresholds. The notion of stability of the extremum graph proves effective to finding suitable threshold values. In certain cases, no obvious stability regions are found. Again, interactivity is key and the ability to explore these in real-time minimizes the effort required to find good values. Topological spines are designed to preserve structural properties, including the local symmetries and cardinalities of extrema, and the relative volume of different spatial regions. Other geometric properties, such as shape and curvature, are usually not preserved in our approach. We believe our technique is an abstraction suitable for spatial reasoning that may be combined with other techniques to supply missing geometric information.

7 Summary

When confronted with tasks that involve spatial reasoning, direct visualization of complex, possibly high dimensional datasets proves impractical. Even for 3D data sets, we have shown that contour trees do not preserve the locality of extrema and other critical points.

In this paper, we have presented a data structure—the extremum graph—and a visual representation that augments this structure with geometric information—the topological spine. After our study of this representation and validation on a number of scalar fields, we have identified a number of practical use cases for this representation: For 3D scalar field visualization, the topological spine provides an overview of the data set as a network of 1D embedded structures that are usually sparse. This approach solves the occlusion issues of direct volume rendering and the clutter associated with projections. Although 3D exploration is still an important task, the topological spine provides an occlusion-free 2D map that helps identify structures faster and more accurately. For higher-dimensional exploration, there is no direct counterpart to 3D volume rendering, and one usually resorts to projections. Because the topological spine encodes geometric information, it can suggest to the user where to steer their attention.

Topological spines create shape signatures in 2D to represent complex relationships between extrema. In our studies, we have encountered that patterns such as hierarchies of extrema, cycles (e.g., tori in 3D), and fractals, can be identified with ease using our representation. This proves useful for understanding complex structures by analyzing the formation of structures that are smaller and easier to comprehend.

As shown in our examples and the accompanying video, interactivity plays an important role in understanding structure. Here, interactivity refers to the ability to explore the structural parameters of a scalar function, in terms of what defines noise vs. a feature, and how well connected features are. We introduced a parameter space as a range in persistence, which extends the idea of topological simplification towards the preservation of structurally important connections between extrema. We found that the notion of stability of persistence is a valuable hint that aids in the selection of parameter values.

Our approach defines structure in terms of low-level parameters that govern the topology of a scalar field. Higher-level properties, such as symmetry, appear as a side-effect. Alternatively, one can formulate the problem of structure-preserving visualization as the generation of an image that best depicts a target structure. Our approach currently employs local strategies, but topological spines can also leverage global strategies, including automatic symmetry detection.

REFERENCES

[1] D. H. Ackley. *A connectionist machine for genetic hillclimbing*. Kluwer, Boston, 1987.

[2] R. F. W. Bader. *Atoms in Molecules: A Quantum Theory*. Oxford University Press, 1994.

[3] K. Beketayev, G. Weber, M. Haranczyk, M. Hlawitschka, P.-T. Bremer, and B. Hamann. Topology-based visualization of transformation pathways in complex chemical systems. In *Eurographics Symposium on Visualization*, 2011. To appear.

[4] P.-T. Bremer, V. Pascucci, and B. Hamann. Maximizing adaptivity in hierarchical topological models. In *Shape Modeling and Applications*, pages 298–307, 2005.

[5] P.-T. Bremer, G. Weber, J. Tierny, V. Pascucci, M. Day, and J. B. Bell. Interactive exploration and analysis of large scale simulations using topology-based data segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 2010. To appear.

[6] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.

[7] H. Carr, J. Snoeyink, and M. van de Panne. Flexible isosurfaces: Simplifying and displaying scalar topology using the contour tree. *Computational Geometry*, 43(1):42–58, 2010.

[8] F. Chazal, L. Guibas, S. Oudot, and P. Skraba. Analysis of scalar fields over point cloud data. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1021–1030, 2009.

[9] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *ACM Symposium on Computational geometry*, pages 361–370, 2003.

[10] H. Edelsbrunner, D. Morozov, and V. Pascucci. Persistence-sensitive simplification functions on 2-manifolds. In *Proceedings of the twenty-second annual symposium on Computational geometry*, SCG '06, pages 127–134, New York, NY, USA, 2006. ACM.

[11] E. Espinosa, M. Souhassou, H. Lachekar, and C. Lecomte. Topological analysis of the electron density in hydrogen bonds. *Acta Crystallographica Section B-structural Science*, 55:563–572, 1999.

[12] I. Fujishiro, Y. Takeshima, T. Azuma, and S. Takahashi. Volume data mining using 3D field topology analysis. *IEEE Computer Graphics & Applications*, 20(5):46–51, 2000.

[13] S. Gerber, P.-T. Bremer, V. Pascucci, and R. T. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1271–1280, 2010.

[14] A. Gyulassy, P.-T. Bremer, V. Pascucci, and B. Hamann. A practical approach to Morse-Smale complex computation: Scalability and generality. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1619–1626, 2008.

[15] A. Gyulassy, V. Natarajan, V. Pascucci, P.-T. Bremer, and B. Hamann. Topology-based simplification for feature extraction from 3D scalar fields. *IEEE Transactions on Computer Graphics and Visualization*, 12(4):474–484, 2006.

[16] A. Gyulassy, V. Natarajan, V. Pascucci, and B. Hamann. Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1440–1447, 2007.

[17] W. Harvey and Y. Wang. Topological landscape ensembles for visualization of scalar-valued functions. *Computer Graphics Forum*, 29(3):993–1002, 2010.

[18] J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics & Applications*, 11(3):36–46, 1991.

[19] C. K. Johnson, M. N. Burnett, and W. D. Dunbar. *Crystallographic Topology and its Applications*, pages 1–25. 1996.

[20] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1052–1060, 2006.

[21] L. Leherte, J. Glasgow, K. Baxter, E. Steeg, and S. Fortier. Analysis of Three-Dimensional Protein Images. *Journal of Artificial Intelligence Research*, 7:125–159, 1997.

[22] P. Oesterling, C. Heine, H. Janicke, G. Scheuermann, and G. Heyer. Visualization of high dimensional point clouds using their density distribution's topology. *IEEE Transactions on Visualization and Computer Graphics*, 2011. To appear.

[23] V. Pascucci, K. Cole-MacLaughlin, and G.Scorzelli. Multi-resolution computation and presentation of contour trees. Technical Report UCRL-PROC-208680, Lawrence Livermore National Laboratory, 2005.

[24] V. Pascucci and K. Cole-McLaughlin. Parallel computation of the topology of level sets. *Algorithmica*, 38:249–268, 2003.

[25] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: Simplicity and speed. *ACM Transactions on Graphics*, 26(3):58.1–58.9, 2007.

[26] J. Sahner, B. Weber, S. Prohaska, and H. Lamecker. Extraction of feature lines on surface meshes based on discrete Morse theory. *Computer Graphics Forum*, 27(3):735–742, 2008.

[27] H. Schwefel. *Numerical optimization of computer models*. Wiley, 1981.

[28] Y. Shinagawa and T. Kunii. Constructing a Reeb graph automatically from cross sections. *IEEE Computer Graphics & Applications*, 11(5):44–51, 1991.

[29] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):24–49, 2004.

[30] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P.Seidel. Saddle connectors — An approach to visualizing the topological skeleton of complex 3D vector fields. In *IEEE Visualization*, pages 225–232, 2003.

[31] J. Tierny, A. Gyulassy, E. Simon, and V. Pascucci. Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1177–1184, 2009.

[32] I. G. Tollis, G. Di Battista, P. Eades, and R. Tamassia. *Graph Drawing: Algorithms for the Visualization of Graphs*. Prentice Hall, July 1998.

[33] M. J. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *ACM Symposium on Computational Geometry*, pages 212–220, 1997.

[34] G. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1077–2626, 2007.

[35] G. Weber, G. Scheuermann, H. Hagen, and B. Hamann. Exploring scalar fields using critical isovalues. In *IEEE Visualization*, pages 171–178, 2002.

[36] T. Weinkauf and D. Günther. Separatrix persistence: Extraction of salient edges on surfaces using topological methods. *Computer Graphics Forum*, 28(5):1519–1528, 2009.