

A Topological Hierarchy for Functions on Triangulated Surfaces

Peer-Timo Bremer^{*‡}, *Member, IEEE*, Herbert Edelsbrunner[†], Bernd Hamann^{*}, *Member, IEEE*, and Valerio Pascucci[‡], *Member, IEEE*,

Abstract—We combine topological and geometric methods to construct a multi-resolution representation for functions over two-dimensional domains. In a preprocessing stage, we create the Morse-Smale complex of the function and progressively simplify its topology by canceling pairs of critical points. Based on a simple notion of dependency among these cancellations we construct a hierarchical data structure supporting traversal and reconstruction operations similarly to traditional geometry-based representations. We use this data structure to extract topologically valid approximations that satisfy error bounds provided at run-time.

Index Terms—Critical point theory, Morse-Smale complex, terrain data, simplification, multi-resolution data structure.

I. INTRODUCTION

THE efficient construction of topologically and geometrically simplified models is a central problem in visualization. This paper describes a hierarchical data structure representing the topology of a continuous function on a triangulated surface. An example of such data is the distribution of the electrostatic potential on a molecular surface or elevation data on a sphere (e.g., the Earth). The complete topology of the function is computed and encoded in a hierarchy that provides fast and consistent access to adaptive topological simplifications. Additionally, the hierarchy includes geometrically consistent approximations of the function corresponding to any topological refinement. In the special case of a planar domain, the function can be thought of as elevation and the graph of the function as a surface in three-dimensional space. In this case our framework creates a topology-based hierarchy of the geometry of this surface.

A. Motivation

Scientific data often consists of measurements over a geometric domain or space. We can think of it as a discrete sample of a continuous function over the space. We are interested in the case in which the space is a triangulated surface (with or without boundary).

A hierarchical representation is crucial for efficient and preferably interactive exploration of scientific data. The traditional approach to constructing such a representation is based on progressive data simplification driven by a numerical measurement of the error. Alternatively, we may drive the simplification process with measurements of topological features.

Such an approach is appropriate if topological features and their spatial relationships are more essential than geometric error bounds to understand the phenomena under investigation. An example is water flow over a terrain, which is influenced by possibly subtle slopes. Small but critical changes in elevation may result in catastrophic changes in water flow and accumulation. Thus, our approach is distinctly different from one that is purely driven by numerical approximation error. It ensures that topology of a function is preserved as long as possible during a simplification process, which is not necessarily the case with simplification methods driven by approximation error.

There are applications beyond the analysis of measured data. For example, we may artificially create a continuous function over a surface and use that function to guide the segmentation of the surface into patches.

B. Related work

The topological analysis of scalar valued scientific data has been a long standing research focus. Morse-theory-related methods have already been developed in the 19th century [1], [2], long before Morse theory itself was formulated, and hierarchical representations have been proposed [3], [4] without making use of the mathematical framework developed by Morse and others [5], [6]. However, most of this research was lost and has been rediscovered only recently. Most modern research in the area of multi-resolution structures is geometric and many techniques have been developed during the last decade. The most successful algorithms developed in that era are based on edge contraction as the fundamental simplifying operation [7], [8] and accumulated square distances to plane constraints as the error measure [9], [10]. This work focused on triangulated surfaces embedded in three-dimensional Euclidean space, which we denote as \mathbb{R}^3 . We find a similar focus in the successive attempts to include the capability to change the topological type [11], [12].

In the field of flow visualization topological analysis and topology based simplification are based on the work by Helman and Hesselink [13]. They proposed a structure similar to the Morse complex to analysis vector fields and later methods to simplify this complex have been developed [14], [15], [16]. Unfortunately, computing such a complex relies on numerical integration along inherently unstable regions of the vector field and is therefore limited to relatively small and clean data sets. For the simpler case of piece-wise linear scalar valued functions (whose gradients define a piece-wise constant flow-field) we compute the topology in a symbolic manner which is robust even in degenerate cases. Therefore, we can compute Morse complexes for data sets with tens of thousands of

^{*}Center for Image Processing and Integrated Computing, Department of Computer Science, University of California, Davis, CA 95616-8562

[†]Department of Computer Science, Duke University, Durham, NC 27708 and Raindrop Geomagic, Research Triangle Park, NC 27709

[‡]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551

critical points compared to hundreds of critical points in commonly used vector fields [14], [15], [16]. Unlike the method in [14] we maintain a consistent geometric approximation of the topology and do not create higher-order criticalities as it is done in [15]. Additionally, our error bound is directly linked to the approximation error, see Section V-A, and we provide a multi-resolution hierarchy rather than a simplification strategy.

To remove (spurious) topological features from all level sets simultaneously, we interpret the critical points of the function as the culprits responsible for topological features that appear in the level sets [17], [18]. While sweeping through the level sets we see that critical points indeed start and end such features, and we may use the length of the interval over which a feature exists as a measure of its importance. For the special case of two-dimensional height fields this measure was first proposed by Horman [19] and later adopted by Mark [20]. We use the more general concept of persistence introduced in [21], where the Morse-Smale complex of the function domain occupies a central position. Its construction and simplification is studied for 2-manifolds in [22] and for 3-manifolds in [23].

C. Results

We follow the approach taken in [22], with some crucial differences and extensions. Given a piecewise linear continuous function over a triangulated domain, we

1. construct a decomposition of the domain into monotonic quadrangular regions by connecting critical points with lines of steepest descent;
2. simplify the decomposition by performing a sequence of cancellations ordered by persistence; and
3. turn the simplification process into the construction of a hierarchical multi-resolution data structure whose levels correspond to simplified versions of the function.

The first two steps are discussed in [22], but the third step is new. Nevertheless, this paper makes original contributions to all three steps and in the application of the data structure to concrete scientific problems. These contributions are

- (i) a modification of the algorithm of [22] that constructs the Morse-Smale complex without the use of handle slides;
- (ii) the simplification of the complex by simultaneous application of independent cancellations;
- (iii) a numerical algorithm to approximate the simplified function;
- (iv) a shallow multi-resolution data structure combining the simplified versions of the function into a single hierarchy;
- (v) an algorithm for traversing the data structure that combines different levels of the hierarchy to construct adaptive simplifications; and
- (vi) the application of our method to various data sets.

The hallmark of our method is the fusion of the geometric and topological approaches to multi-resolution representations. The entire process is controlled by topological considerations, and the geometric method is used to realize monotonic paths and patches. The latter plays a crucial but sub-ordinate role in the overall algorithm.

II. BACKGROUND

We describe an essentially combinatorial algorithm based on intuitions provided by investigations of smooth maps. In this section, we describe the necessary background, in Morse theory [6], [24] and in combinatorial topology [25], [26].

A. Morse functions

Throughout this paper, \mathbb{M} denotes a compact 2-manifold without boundary and $f: \mathbb{M} \rightarrow \mathbb{R}$ denotes a real-valued smooth function over \mathbb{M} . Assuming a local coordinate system at a point $a \in \mathbb{M}$, we compute two partial derivatives and call a *critical* when both are zero and *regular* otherwise. Examples of critical points are maxima (f decreases in all directions), minima (f increases in all directions), and saddles (f switches between decreasing and increasing four times around the point).

Using the local coordinates at a , we compute the *Hessian* of f , which is the matrix of second partial derivatives. A critical point is *non-degenerate* when the Hessian is non-singular, which is a property that is independent of the coordinate system. According to the Morse Lemma, it is possible to construct a local coordinate system such that f has the form $f(x_1, x_2) = f(a) \pm x_1^2 \pm x_2^2$ in a neighborhood of a non-degenerate critical point. The number of minus signs is the *index* of a and distinguishes the different types of critical points: minima have index 0, saddles have index 1, and maxima have index 2. Technically, f is a *Morse function* when all its critical points are non-degenerate and have pairwise different function values. Most of the challenges in our method are rooted in the need to enforce these conditions for given functions that do not satisfy them originally.

B. Morse-Smale complexes

Assuming a Riemannian metric and an orthonormal local coordinate system, the *gradient* at a point a of the manifold is the vector of partial derivatives. The gradient of f forms a smooth vector field on \mathbb{M} , with zeroes at the critical points. At any regular point we have a non-zero gradient vector, and when we follow that vector we trace out an *integral line*, which starts at a critical point and ends at a critical point while technically not containing either of them. Since integral lines ascend monotonically, the two endpoints cannot be the same. Because f is smooth, two integral lines are either disjoint or the same. The set of integral lines covers the entire manifold, except for the critical points. The *descending manifold* $D(a)$ of a critical point a is the set of points that flow toward a . More formally, it is the union of a and all integral lines that end at a . For example, the descending manifold of a maximum is an open disk, that of a saddle is an open interval, and that of a minimum is the point itself. The collection of stable manifolds is a complex, in the sense that the boundary of a cell is the union of lower-dimensional cells. Symmetrically, we define the *ascending manifold* $A(a)$ of a as the union of a and all integral lines that start at a .

For the next definition, we need an additional non-degeneracy condition, namely that ascending and descending

manifolds that intersect do so transversally. For example, if an ascending 1-manifold intersects a descending 1-manifold then they cross. Due to the disjointness of integral lines, this implies that the crossing is a single point, namely the saddle common to both. Assuming that this transversality property is satisfied, we overlay the two complexes and obtain what we call the *Morse-Smale complex*, or MS complex, of f . Its cells are the connected components of the intersections between ascending and descending manifolds. Its vertices are the vertices of the two overlaid complexes, which are the minima and maxima of f , together with the crossing points of ascending and descending 1-manifolds, which are the saddles of f . Each 1-manifold is split at its saddle, thus contributing two arcs to the MS complex. Each saddle is endpoint of four arcs, which alternately ascend and descend around the saddle. Finally, each region has four sides, namely two arcs emanating from a minimum and ending at two saddles and two additional arcs continuing from the saddles to a common maximum. It is generically possible that the two saddles are the same, in which case two of the four arcs merge into one. The region lies on both sides of the merged arc so it makes sense to double-count and to maintain that the region has four sides. An example is shown in Fig. 1.

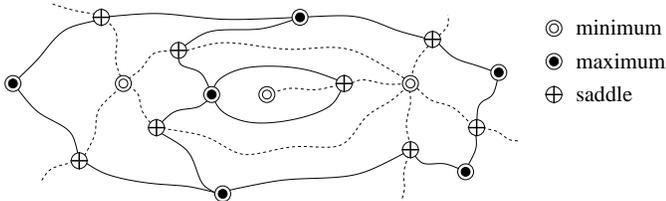


Fig. 1. A sample MS complex.

C. Piecewise linear functions

Functions occurring in scientific applications are rarely smooth and mostly known only at a finite set of points spread out over a manifold. It is convenient to assume that the function has pairwise different values at these points. We assume that the points are the vertices of a triangulation K of \mathbb{M} , and we extend the function values by piecewise linear interpolation applied to the edges and triangles of K . The *star* of a vertex u consists of all simplices (vertices, edges and triangles) that contain u , and the *link* consists of all faces of simplices in the star that are disjoint from u . Since the surface defined by K is a 2-manifold, the link of every vertex is a topological circle. The *lower star* contains all simplices in the star for which u is the highest vertex, and the *lower link* contains all simplices in the link whose endpoints are lower than u . Note that the lower link is the subset of simplices in the link that are faces of simplices in the lower star. Topologically, the lower link is a subset of a circle. We define what we mean by a critical point of a piecewise linear function based on the lower link. As illustrated in Fig. 2, the lower link of a *maximum* is the entire link and that of a *minimum* is empty. In all other cases, the lower link of u consists of $k + 1 \geq 1$ connected pieces, each being an arc or possibly a single vertex.

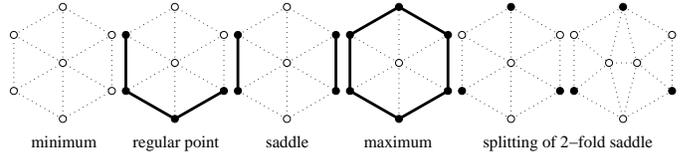


Fig. 2. Classification of a vertex based on relative height of vertices in its link. The lower link is marked black.

The vertex u is *regular* if $k = 0$ and a *k -fold saddle* if $k \geq 1$. As illustrated in Fig. 2 for $k = 2$, a k -fold saddle can be split into k simple or 1-fold saddles.

D. Persistence

We require a numerical measure of the importance of critical points that can be used to drive the simplification of a MS complex. For this purpose, we pair up critical points and use the absolute difference between their heights as importance measure. To construct the critical point pairs, we imagine sweeping the 2-manifold \mathbb{M} in the direction of increasing height. This view is equivalent to sorting the vertices by height and incrementally constructing the triangulation K of \mathbb{M} one lower star at a time. The topology of the partial triangulation changes whenever we add a critical vertex, and it remains unchanged whenever we add a regular vertex. Except for some exceptional cases that have to do with the surface type of \mathbb{M} , each change either *creates* a component or an annulus or it *destroys* a component (by merging two) or an annulus (by filling the hole). We pair a vertex v that destroys with the vertex u that created what v destroys. The *persistence* of u and of v is the delay between the two events: $p = f(v) - f(u)$. An algebraic justification of this definition and a fast algorithm for constructing the pairs can be found in [21].

III. MORSE-SMALE COMPLEX

We introduce an algorithm for computing the MS complex of a function f defined over a triangulation K . In particular, we compute the ascending and descending 1-manifolds (paths) of f starting from the saddles, and use them to partition K into quadrangular regions which define the MS complex.

A. Path construction

Starting from each saddle, we construct two lines of steepest ascent and two lines of steepest descent. We do not adopt the original algorithm proposed in [22] and follow actual lines of maximal slope instead of edges of K . In particular, we split triangles to create new edges in the direction of the gradient. We modify this basic strategy to avoid regions with disconnected interior and regions whose interior does not touch both saddles. Without the modification such regions may be created because f is not smooth and integral lines can merge. Fig. 3(a) shows one such case, where paths merge at *junctions* and disconnect the interior of a region into two. The modification that eliminates the two undesired configurations consists of disallowing two paths to merge if they are of different type; see Fig. 3(b). Two paths are still allowed to merge if they are both ascending or both descending. If two paths are not allowed to merge we split one edge of the

triangulation and introduce a new sample with function value that preserves the structure of the MS complex but locally avoids the junction. Fig. 4 shows the repeated application of this strategy to avoid a junction. Practically, this situation rarely occurs and it can be shown that in the worst case the number of triangles introduced is linear in the size of the mesh.

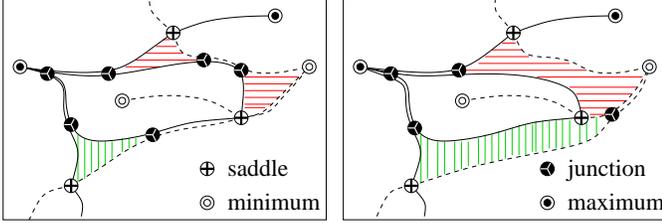


Fig. 3. Portion of the MS complex of a piecewise linear function. Since the gradient is not continuous, (solid) ascending and (dotted) descending paths can meet in junctions and share segments. (Left) Complex with no restrictions on sharing segments. The green region touches only one saddle, and the red one is disconnected. (Right) Only path of the same type can meet. The interior of each region is connected and touches both saddles.

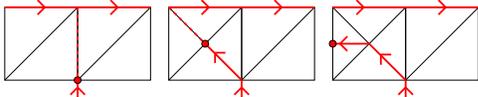


Fig. 4. Triangle split to keep two paths separated. Solid red lines indicate two portions of paths already computed. (Left) The red circle is the current extremum of a path that would follow the red dotted line. (Middle) The path is extended splitting a first triangle. (Right) Since the two paths would still intersect, a second triangle is split.

After computing all paths, we partition K into quadrangular regions forming the cells of the MS complex. Specifically, we grow each quadrangle from a triangle incident to a saddle without ever crossing a path.

In degenerate areas of \mathbb{M} , where several vertices may have the same function value, the greedy choices of local steepest ascent/descent may not work consistently. We address this problem using the *simulation of simplicity*, or (SoS) [27]. We orient each edge of K in the direction of ascending function value. Vertex indices are used to break ties on flat edges such that the resulting directed graph has no cycles. Using these orientations, the search for the steepest path is transformed to a weighted-graph search and function values are only used as preferences. Thus, our algorithm is robust even for highly degenerate data sets as the one shown in Fig. 5.

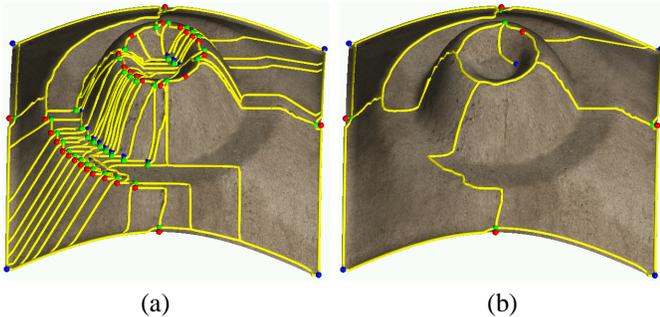


Fig. 5. MS complex of degenerate data set. The “volcano” is created by a $\sin()$ function that is flat both inside the “crater” and at the foot of the mountain. (a) Originally computed MS complex. A large number of critical points is created by eliminating flat regions using simulation of simplicity. (b) The same complex after removal of “topological noise.”

```

Let  $T = \{F, E, V\}$  be the triangulation of  $\mathbb{M}$ ;
initialize the MS complex,  $M = \emptyset$ ;
initialize the sets of paths and cells,  $P = C = \emptyset$ ;
initialize SoS to direct the edges of  $T$ ;
 $S = \text{FINDSADDLES}(T)$ ;
 $S = \text{SPLITMULTIPLESADDLES}(T)$ ;
 $\text{SORTBYHEIGHT}(S)$ ;
forall  $s \in S$  in ascending order do
   $\text{COMPUTEASCENDINGPATH}(P)$ 
endfor;
forall  $s \in S$  in descending order do
   $\text{COMPUTEDESCENDINGPATH}(P)$ 
endfor;
while there exists untouched  $f \in F$  do
   $\text{GROWREGION}(f, p_0, p_1, p_2, p_3)$ ;
   $\text{CREATEMORSECELL}(C, p_0, p_1, p_2, p_3)$ 
endwhile;
 $M = \text{CONNECTMORSECELLS}(C)$ .

```

Fig. 6. Sequence of high-level operations used to create an MS complex. When we grow a cell from a triangle f we encounter the four boundary paths. p_0 to p_3 , which are then incorporated into a half-edge representation of the cell

B. Diagonals and diamonds

The central element of our data structure for the MS complex is the neighborhood of a simple saddle or, equivalently, the halves of the quadrangles that share the saddle as one of their vertices. To be more specific about the halves, recall that in the smooth case each quadrangle consists of integral lines that emanate from its minimum and end at its maximum. Any one of these integral lines can be chosen as *diagonal* to decompose the quadrangle into two triangles. The triangles sharing a given saddle form the *diamond* centered at the saddle. As illustrated in Fig. 8(a), each diamond is a quadrangle whose vertices alternate between minima and maxima around the saddle in its center. It is possible that two vertices are the same and the boundary of the diamond is glued to itself along two consecutive diagonals.

C. The algorithm

We compute the descending paths starting from the highest saddle and the ascending paths starting from the lowest saddle. Thus, when two paths aim for the same extremum, the one with higher persistence (importance) is computed first. The boundary of the data set is artificially tagged as a path. The complete algorithm is summarized in Fig. 6.

IV. HIERARCHY

Our main objective is the design of a hierarchical data structure that supports adaptive coarsening and refinement of the data. In this section, we describe such a data structure and discuss how to use it.

A. Cancellations

We use only one atomic operation to simplify the MS complex of a function, namely a *cancellation* that eliminates two critical points. The inverse operation that creates two critical

points is referred to as an *anti-cancellation*. In order to cancel two critical points they must be adjacent in the MS complex. Only two possible combinations arise: a minimum and a saddle or a saddle and a maximum. The two configurations are symmetric, and we can limit the discussion to the second case, which is illustrated in Fig. 7.

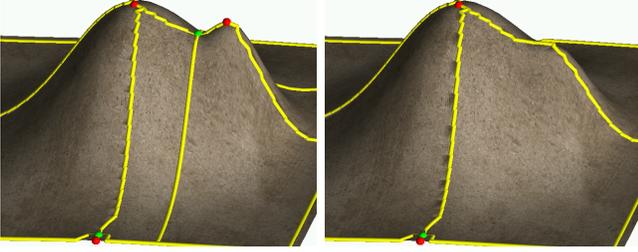


Fig. 7. Portion of the graph of a function before (left) and after (right) cancellation of a maximum (red) and a saddle (green).

Let u be the saddle and v the maximum of the canceled pair, and let w be the other maximum connected to u . We require $w \neq v$ and $f(w) > f(v)$; otherwise, we prohibit the cancellation of u and v . We view the cancellation as merging three critical points into one, namely u, v, w into w . All paths ending at either u, v , or w are removed and we adapt the local geometry to the new topology, as described in Section V. Subsequently, all paths that were connected to either maximum are recomputed. In other words, we connect every saddle on the boundary of the geometrically adapted region to the unique maximum within the region. To avoid excessive splitting of the triangulation we restrict the re-computed paths to share edges of the triangulation. There are several reasons for requiring $f(w) > f(v)$: it implies that all recomputed paths remain monotonic and ensures that we do not eliminate any level sets, except that the ones between $f(u)$ and $f(v)$ are simplified. We may think of a cancellation as deleting the two descending paths of u and contracting the two ascending paths of u .

B. Node removal

We construct the multi-resolution data structure from bottom to top. The bottom layer stores the MS complex of the function f , or, to be more precise, the corresponding decomposition of the 2-manifold into diamonds. Fig. 8(b) illustrates this layer

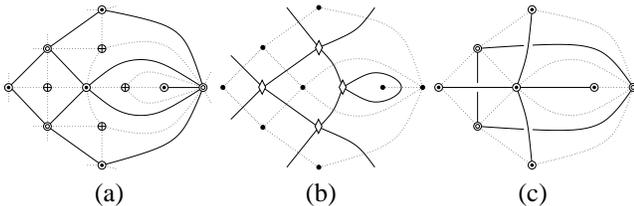


Fig. 8. (a) The (dotted) portion of an MS complex and the (solid) portion of the corresponding decomposition into diamonds. (b) Portion of the data structure (solid) representing the piece of the decomposition into diamonds (dotted). (c) Cancellation graph (solid) of the decomposition into diamonds (dotted).

by showing each diamond as a node with arcs connecting it to neighboring diamonds. Each node has degree four, but there can be loops starting and ending at the same node. A cancellation corresponds to removing a node and re-connecting

its neighbors. When this node is shared by four different arcs we can connect the neighbors in two different ways. As illustrated in Fig. 9, this operation corresponds to the two different cancellations merging the saddle with the two adjacent maxima or the two adjacent minima. There is only one way to remove a node shared by a loop and two other arcs, namely to delete the loop and connect the two neighbors.

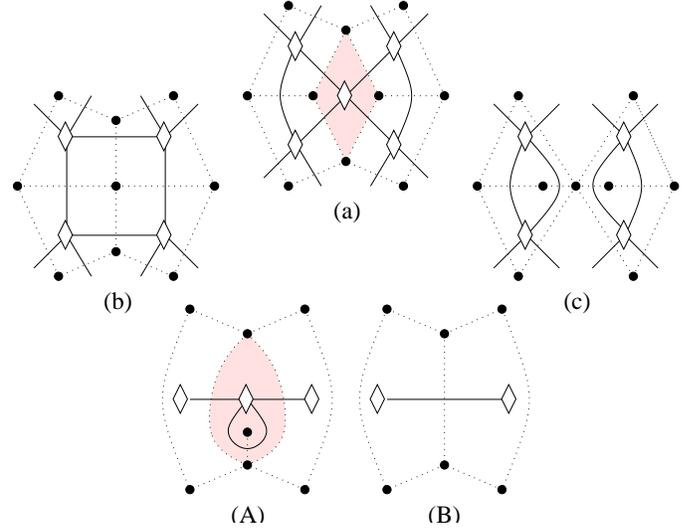


Fig. 9. The four-sided diamond (a) can be zipped up in two ways: from top to bottom (b) or from left to right (c). A folded diamond (A) can be zipped up in only one way (B).

To construct the hierarchy by repeated cancellations, we use the algorithm in [21] to match critical points in pairs $(s_1, v_1), (s_2, v_2), \dots, (s_k, v_k)$, with persistence increasing from left to right. Let Q_j be the MS complex obtained after the first j cancellations, for $0 \leq j \leq k$. We obtain Q_{j+1} by modifying Q_j and storing sufficient information so we can recover Q_j from Q_{j+1} . The hierarchy is complete when we reach Q_k . We call each Q_j a *layer* in the hierarchy and represent it by activating its diamonds as well as neighbor and vertex pointers and de-activating all other diamonds and pointers. To ascend in the hierarchy (coarsen the quadrangulation) we de-activate the diamond of s_{j+1} ; to descend in the hierarchy (refine the quadrangulation) we activate the diamond of s_{j-1} . Activating and de-activating a diamond requires updating of only a constant number of pointers.

C. Independent cancellations

We generalize the notion of a layer in the hierarchy to permit view-dependent simplifications. The key concept here is the possibility to interchanging two cancellations. The most severe limitation to interchanging cancellations derives from the assignment of extrema as vertices of the diamonds and from re-drawing the paths ending at these extrema. To understand this limitation, we introduce the *cancellation graph* whose vertices are the minima and maxima. Fig. 8(c) shows an example of such a graph. For each diamond, there exists an edge connecting the two minima and another edge connecting the two maxima. There are no loops and therefore sometimes only one edge per diamond. Zipping up a diamond corresponds to contracting one of the edges and deleting the other, if it

exists. One endpoint of the edge remains as a vertex and the other disappears, implying that the diamonds that share the second endpoint receive a new vertex. A special case arises when a diamond shares both endpoints: the connecting edge that would turn into a loop is deleted.

Two cancellations in a (possibly simplified) MS complex are *interchangeable* when it is irrelevant in which order the two operations are applied to the data structure. For example, the two cancellations zipping up the same diamond are not interchangeable since one preempts the other. In general, two cancellations are interchangeable when their diamonds share no vertex, a condition we refer to as being *independent*. Note that two interchangeable cancellations are not necessarily independent. Even though independence is the more limiting of the two concepts, it offers sufficient flexibility in choosing layers to support the adaptation of the representation to external constraints, such as the biased view of the data.

When we can perform a relatively large number of independent cancellations we have more freedom generating layers in the multi-resolution data structure. Ideally, we would like to identify a large independent set and iterate to construct a shallow hierarchy. However, in the worst case, every pair of cancellations is dependent, which makes the construction of a shallow hierarchy impossible. As illustrated in Fig. 11(a), such a configuration exists even for the sphere and for any arbitrary number of vertices. Nevertheless, worst-case situations are unlikely to arise as they require a large number of folded diamonds. Specifically, it is possible to prove that every MS complex without folded diamonds implies a linear number of independent cancellations.

V. GEOMETRIC APPROXIMATION

After each cancellation, we create or change the geometry that locally defines f . We pursue three objectives: the approximation must agree with the given topology, the error should be small, and the approximation should be smooth.

A. Error bounds

We measure the error as the difference between function values at a point. It is convenient to think of the graph of f as the geometry and this difference as the (vertical) distance between the original and the simplified geometry at the location of the point. The persistence of the critical points involved in a cancellation implies a lower bound on the local error. We illustrate this connection for the one-dimensional case in Fig. 10(a). Recall that the persistence p of the maximum-minimum pair is the difference in their function values. Any monotonic approximation of the curve between the two critical points has an error of at least $p/2$. We can achieve an error of $p/2$, but only if we accept a flat segment for this portion of the curve, see the red curve in Fig. 10(a). When it is allowed to exceed $p/2$, smoother approximations without flat segments are possible, such as the green curve in the same figure. Note that the above describes only the error between the two functions before and after the one cancellation. The error caused by the composition of two or more cancellations is more difficult to analyze and will not be discussed in this paper.

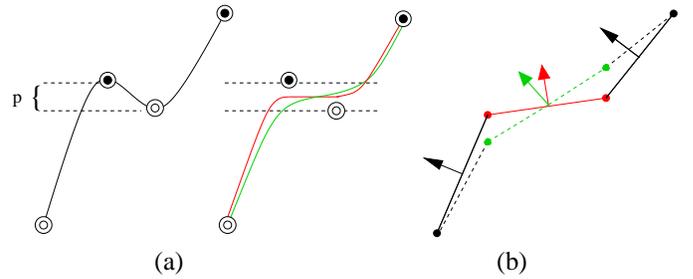


Fig. 10. Geometry fitting for paths: (a) One-dimensional cancellation and several monotonic approximations. (b) Local averaging used to construct smoothly varying monotonic approximations. Slopes of neighboring edges are combined with the original slope, and the function values are adjusted accordingly (edge normals are shown).

B. Data fitting

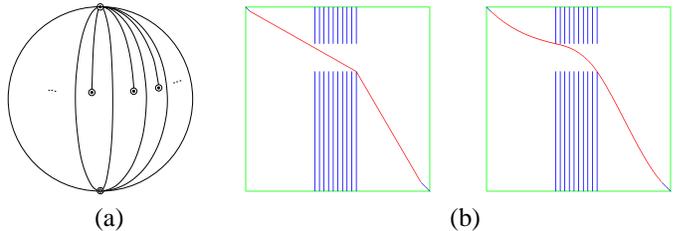


Fig. 11. (a) MS complex on the sphere with pairwise dependent cancellations. (b) One-dimensional smoothing with (blue) error constraints and prescribed endpoint derivatives. (Left) Initial configuration; (right) Constructed solution.

We know that monotonic patches exist, provided we are tolerant to errors. Our goal is therefore to find monotonic patches that minimize some error measure. A large body of literature deals with the more general topic of shape-constrained approximation [28], [29]. The general problem is to construct the smoothest interpolant to a set of input data while observing some shape constraints (e.g., convexity, monotonicity, and boundary conditions). However, most published work uses penalty functions instead of tight error bounds. Additionally, the techniques are typically described for tensor product setting, and the definitions of monotonicity for the bivariate case vary and differ from the one we use.

We did not adapt standard techniques for our purposes. Instead, we decided to use a multi-stage iterative approach to construct the geometry that specifies the simplified representation of f . It provides a smooth C^1 -continuous approximation within a specified error bound along the boundaries of the quadrangular patches and a similar approximation but without observing an error bound in the interior of the patches. The paths are constructed iteratively by smoothing the gradients along the edges and post-fitting the function values, as illustrated in Fig. 10(b). During each iteration, we first compute the new gradient of an edge as a convex combination of its gradient and the gradients of the adjacent edges. We then adjust the function values at the vertices to realize the new gradients. During an iteration, we maintain the error bound at the vertices and make sure that the completed path is monotonic. In addition, the gradient at the critical points is set to zero.

The technique performs well in practice although it converges slowly. Sample results are shown in Fig. 11(b). The

interior of the quadrangular patches are modified by applying standard Laplacian smoothing to the function values [30]. During each iteration, the value at a vertex is averaged with those of its neighbors. Since the boundaries are monotonic, this procedure converges to a monotonic solution for the patch interior. We summarize the steps of the geometry fitting process:

1. Find all paths affected by a cancellation;
2. use the gradient smoothing to geometrically remove the canceled critical points;
3. smooth the old regions until they are monotonic;
4. erase the paths and re-compute new paths using the new geometry;
5. use one-dimensional gradient smoothing to force the new paths to comply with the constraints; and
6. smooth the new regions until all points are regular.

The reason for repeating gradient smoothing in Step 5 is this one: The paths constructed in Step 4 are not guaranteed to satisfy the required error bounds. In practice, we do not smooth until the geometry converges, instead perform a predefined number of steps or smooth until all constrains (monotonicity and error bounds) are met (whichever takes longer). Based on our experiments, the constrains are typically easily satisfied in which case a constant number of smoothing steps is performed after each cancellation. Currently, the geometric fitting is the computational bottleneck of the algorithm. However, we are not aware of faster methods to create topologically correct approximations.

VI. REMESHING

While traversing the hierarchy we want to interactively display geometry that agrees with the current topology of the graph of f . Thus, we must determine a triangular mesh within each quadrangular region. The triangulation for each region should be independent of all other regions to allow us to use topology other than that encountered during the creation of the hierarchy.

A. Path smoothing

We first examine the geometric data we want to approximate. Without modifications the algorithms used to compute paths tend to create “non-smooth” paths, see Fig. 12(a). These are visually not pleasing and difficult to approximate. Therefore, we modify the data slightly in order to create smoother paths. We smooth path vertices, except the critical points, in \mathbb{M} (in the case of height fields the xy -plane) without flipping triangles, while preserving their function value. Again, we use Laplacian smoothing that is modified at junctions, see Fig. 12(b). Here, we first average all incoming vertices (based on the direction of the paths) and all outgoing vertices separately. The averages are used to update the junction. This strategy reduces the change in direction between the incoming and outgoing edges rather than minimizing the change of directions between all edges. The result is a more “flow-like” structure, as shown in Fig. 12(c). No vertex can leave its original triangle strip, and, assuming a sufficiently dense base mesh, the overall change in position is minor and critical

points are never moved. In practice, one or two iterations are sufficient to significantly improve path shape.

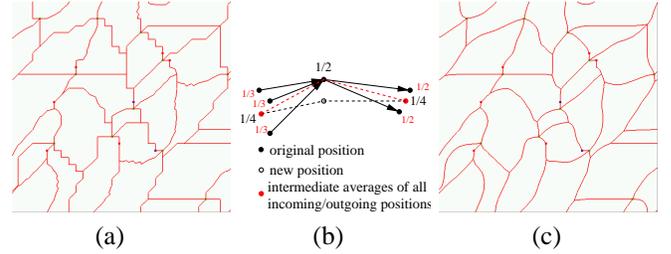


Fig. 12. Path smoothing: (a) A typical path structure without smoothing. (b) Smoothing applied at junctions. All incoming and outgoing vertices are averaged separately. The averages are used to update the position of the junction. (c) The path structure of (a) after two smoothing steps.

B. Parametrization

To enable fast and versatile rendering of the data and reduce memory requirements we remesh each quadrangular region using a regular structure. For this purpose, we use mean-value coordinates as proposed in [31]. We map the boundary of the region to the boundary of one or multiple unit squares. In the interior of the region (which, at this point, is represented as a portion of the triangulation K), we use the fact that each vertex can be expressed as a convex combination of its neighbors. The coefficients in this combination can be computed by solving a sparse linear system. Given the parametrization on the boundary, we use these coefficients to map interior vertices to the parameter space, thus completing the parametrization.

Next, we sample the parameter space on a uniform grid and use its preimage on \mathbb{M} as a new mesh for the region. The grid resolution is chosen based on a given error bound evaluated along boundary paths, which, by construction, follow the direction of maximum change in function value. Specifically, we refine each path until it satisfies the error bound and choose the grid resolution to match the maximum resolution along the four boundary paths.

C. Boundary parametrization

The parameter values of the interior vertices are uniquely defined by the parameter values assigned to the boundary vertices. Therefore, the overall quality of the parametrization relies on a favorable boundary parametrization. The boundary of a region consists of critical points, junctions, and standard path vertices. Independently of the current approximation, the triangulation of a region must always contain its critical points and junctions. The critical points represent the extremal function values of a region and therefore carry maximal information. Junctions are created when two paths that flow toward the same extremum merge. Therefore, each junction replaces a critical point for the region sharing both these paths. To avoid cracks in the mesh all adjacent regions must contain the junction as well.

To remesh the path segments between these extraordinary points (minima maxima, saddles, and junctions) we apply midpoint subdivision based on arclength. To further avoid resolution dependencies between meshes of different regions

we permit T-junctions (hanging nodes) along boundaries. In other words, our representation is not a globally conforming triangulation of \mathbb{M} but rather a collection of patches. Each patch is triangulated with a regular, conforming mesh. We call the collection *crack-free* when the meshes agree geometrically along boundaries. Nevertheless, pixel-wide cracks may appear during rendering as polygons are rasterized at fixed precision. A possible solution is to “fill-in” the cracks during rendering as described by Balázs et al. [32]. Unfortunately, we cannot use continuous surface representations that allow T-junctions such as the method in [33], since they cannot be guaranteed to preserve the topology.

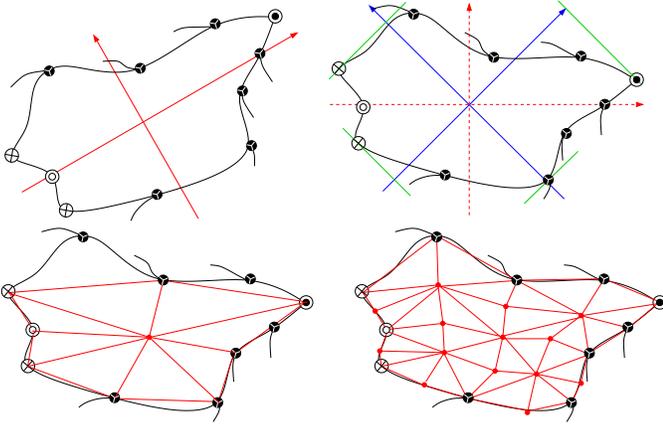


Fig. 13. Creating a parametrization for the boundary. (Top-left) Original region and local coordinate system defined by PCA of all boundary vertices. (Top-right) The vertices are transformed into the PCA coordinate system. In this example, we use a single unit square as base shape in parameter space. The corners are defined by the maximal projected length onto the lines $y = x$ and $y = -x$. (Bottom-left) The regular mesh after the first level of recursively fitting the junctions. (Bottom-right) Final remesh.

What is left to define are the parameter values for special points and the base shape in parameter space. An example is shown in Fig. 13. We perform a principal component analysis (PCA) step in xy -space using all boundary vertices. The distribution ratio between the two principal directions defines the number of consecutive unit squares we use as parameter domain. For a ration of 1 : 2 we use two unit square, for 1 : 3 three etc. Next, we transform the coordinates of all boundary vertices into the PCA coordinate frame using their centroid as origin and the two (normalized) PCA eigenvectors as basis vectors. In the case of a single unit square as parameter space, we compute for all extraordinary vertices $v_i = (x_i, y_i)$ (in the PCA coordinate frame) the scalar product $p_i = (x_i, y_i) * (-1, -1)$. (For two unit squares as base shape we would use $(-2, -1)$ etc..) We assign the lower-left corner in parameter space (parameter value $P(0,0)$) to the vertex v_{LL} with maximal projected length p_i : $LL = \max_i(p_i)$, $P(v_{LL}) = (0,0)$. Similarly, we assign the other three corners in parameter space using projections onto $(1,1)$, $(1,-1)$, and $(-1,1)$. However, we guarantee that all corners are mapped to different vertices. The remaining special vertices are recursively fitted using arc-length parametrization. Once the parameter values for all special vertices are known the parameter values of the path vertices are assigned using their arc-length values.

D. Data layout and rendering

Rather than storing a mesh for each quadrangle explicitly, we use regular grids. This approach allows us to use methods like the one described in [34] for rendering purposes. By storing each grid in what Lindstrom and Pascucci called interleaved embedded quadtrees we avoid having to store connectivity information, while maintaining high flexibility during rendering. This framework can be extended easily to adaptive, view-dependent rendering, as well as efficient view frustum culling and geomorphing. One disadvantage of this data layout is a 33% memory overhead. Another important aspect is the definition of local error coefficients. As we are working with many smaller grids, rather than a single high-resolution one, we must ensure a consistent rendering across boundaries. Since we enforce that samples on grid boundaries are shared their respective error terms agree. Independently of the error term, a region must always render all its junctions which we guarantee by setting their errors to infinity.

VII. RESULTS

We have applied our algorithm to several data sets including terrain data converted from digital elevation models¹, two-dimensional simulation data, and isosurfaces from various scientific data sets. The Puget Sound data set with a resolution of 1025-by-1025 is represented by two-byte integers. The other three terrain data sets are the Needles, Yakima, and Dalles (see Fig. 14) data sets, all of 1201-by-1201 resolution with single-byte integer height values. We have also used simulation data of the autoignition of a spatially non-homogeneous hydrogen-air mixture, courtesy of Echecki and Chen [35], at resolution 512-by-512 with temperature values represented by single-byte unsigned integer values. Additionally, we applied our techniques to three scientific data sets with multiple scalar fields. In particular, we use as manifold domain \mathbb{M} an isosurface of one scalar field and as function f the values of the second field on \mathbb{M} . The first data set, see Fig. 20, represents a methane molecule with \mathbb{M} being an isosurface of the electrostatic potential and f the corresponding van der Waals energy. The second data set, see Fig. 21, describes the interaction energy between a ligand (glucose) and a receptor (ethane) under the three translational degrees of freedom. The domain \mathbb{M} is also an isosurface of the electrostatic potential with f being the van der Waals energy. The third example, see Fig. 22, shows a ground remediation process after an oil spill contamination. The domain \mathbb{M} is an isosurface of the oil concentration in the soil (ground level at the top). The superimposed pseudo-colored function shows the concentration of microbes consuming the oil and performing the remediation process.

The most basic application of our algorithm is removal of topological noise without smoothing. This functionality does not depend on the hierarchy and is implemented by repeated cancellation of critical points with lowest persistence. Our experience suggests that this noise removal step should always be applied at least to remove the artifacts caused by symbolic perturbation. We define all features with persistence

¹<http://www.webgis.com>

below 0.1% of the total data range as noise. Fig. 14 illustrates this procedure for The Dalles data set. Removing the noise reduces the number of critical points from 24,617 to 2,144. As one of the main problem in topological data analysis is the large number of spurious topological features this (symbolic) clean-up is a valuable pre-processing step for many techniques proposed in recent years.

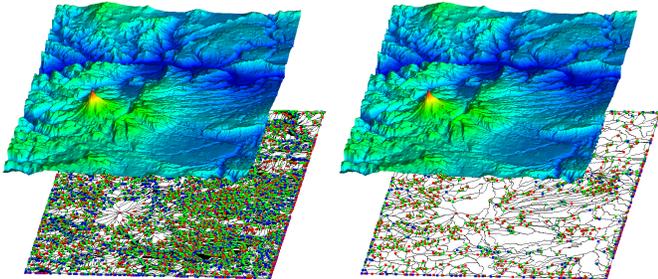


Fig. 14. (Left) Original The Dalles data set containing 24,617 critical points. (Right) Same data with 2,144 critical points after removing all topological features with persistence less than 0.1% of height range.

We have tested several strategies for creating the hierarchy. The first strategy is based on performing cancellations in order of increasing persistence. The second strategy is based on performing simplification in “batches” of maximal independent sets of cancellations. Each batch is created by canceling iteratively the critical point pair with smallest persistence and marking any dependent cancellation as “not-allowed.” Once no more cancellations can be performed all cancellations are reset to be “permitted” and the process is repeated.

The third strategy is a hybrid method of the first two, which builds maximal independent sets with a restricted range of persistence values. In particular, we construct the independent sets as usual but stop if the next cancellation has persistence larger than twice of the smallest in the same set. In all three methods we never remove topology with persistence larger than 20% of the initial height range.

Table I summarizes statistics for the hierarchies constructed for our terrain data. As expected, a hierarchy becomes more shallow when canceling critical points in independent sets. One problem is the presence of nodes with a high number of parents and children; this problem is the result of high-valence vertices in the MS complex. Fig. 15(a) shows the highest-resolution MS complex of the Needles data set, where each path is represented by a straight line from the saddle to the extremum to better highlight the problem. As the data set contains few minima each of them is incident to a very large number of edges. Any cancellation involving one such minimum has a large number of parents and children. High-valence vertices are the cause for the uneven distribution of nodes over the levels.

In Fig. 15(b), we show the number of nodes per level for each cancellation strategy for the Puget Sound data set. The maximum independent set strategy clearly produces superior results in terms of overall shape of the hierarchy. However, this observation does not necessarily imply better practical behavior. Fig. 15(c) shows the number of critical points in the MS complex depending on a uniform error. Equivalent graphs

TABLE I

HIERARCHY STATISTICS FOR DIFFERENT CANCELLATION STRATEGIES. MAXIMAL AND AVERAGE DEPTH (DISTANCE FROM THE ROOT) ARE SHOWN IN THE FIRST TWO COLUMNS. THE LAST THREE COLUMNS LIST THE MAXIMUM NUMBER OF PARENTS AND CHILDREN AND THE AVERAGE DEGREE FOR A NODE.

	depth	avg dep	max #p	max #c	avg deg
Puget Sound	org. no. of CPs 49185, no. of significant CPs 17470				
pure persistence	381	128	148	110	3.80
max. indep. set	157	118	131	112	4.28
indep. set-cut-off	238	105	147	106	3.94
Yakima	org. no. of CPs 21275, no. of significant CPs 6082				
pure persistence	119	56	73	32	3.60
max. indep. set	57	35	38	34	4.24
indep. set-cut-off	96	52	74	37	3.82
Dalles	org. no. of CPs 24617, no. of significant CPs 2144				
pure persistence	80	34	75	39	3.40
max. indep. set	54	31	82	43	3.88
indep. set-cut-off	73	33	63	57	3.52
Needles	org. no. of CPs 17375, no. of significant CPs 3772				
pure persistence	177	68	111	87	3.60
max. indep. set	113	70	87	87	3.88
indep. set-cut-off	149	62	124	101	3.68

for the other data sets are shown in Figs. 16 and 17. Even though the hierarchy created by maximal independent sets is the most shallow it produces a significantly denser mesh. The hybrid method produces nearly identical results during traversal as the one based on pure persistence. However, the hybrid method creates a more shallow hierarchy and a higher branching factor and it is therefore the method of choice.

Fig. 18 shows the Puget Sound data set. The original topology has 49,185 critical points and is too dense to be printed. The upper-left picture shows the data with 4,045 critical points obtained after removing the topological noise using a persistence threshold of 0.5% of the elevation range. The upper-right picture shows the approximation of the data with 2,025 critical points obtained by increasing the persistence threshold to 1.2%. The lower-left picture shows a significantly coarser MS complex containing 289 critical points that remain after increasing the persistence threshold to 20%. The differences between the resolutions are most notable along the front-facing boundary. The pre-processing of the Puget Sound data was done in about three and a half hours, due to the slow convergence of the geometric fitting procedure. The traversal of the hierarchy and rendering are fully interactive.

The hierarchy also supports adaptive refinements, two examples are shown in Fig. 19. Fig. 19(a) shows a view-dependent refinement of the Puget Sound data. The full resolution is preserved inside the view frustum yielding a total of 1,070 critical points. Outside the view frustum, we have simplified the data to the extent possible. One can observe a quick drop in resolution away from the view frustum. Reducing the topology outside the frustum naturally reduces the number of quadrangular regions that must be rendered (and therefore the number of triangles) and frustum culling can be performed directly on the regions culling large parts of the mesh without traversal. Fig. 19(c) shows the combustion data adaptively refined to preserve only maxima with a high function value. Only maxima above 90% of the maximal function value and their ancestors in the hierarchy are preserved. Notice that several lower maxima have completely disappeared.

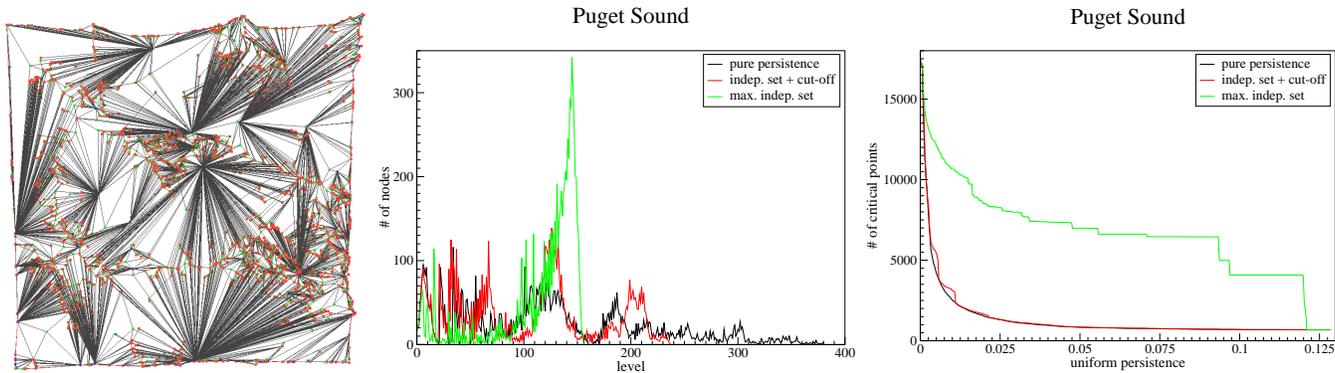


Fig. 15. (left) Highest resolution MS complex of Needles data set.(middle) Node distribution over the levels for different cancellation strategies for Puget Sound data set. (right) Number of critical points in MS complex during uniform refinement of Puget Sound data set.

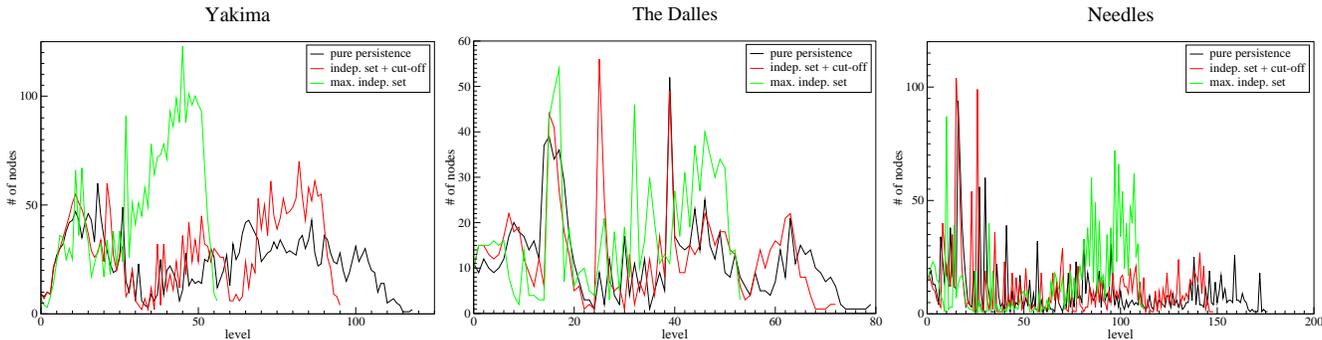


Fig. 16. Nodes distribution over the levels for different cancellation strategies for the Yakima (left), The Dalles (middle), and the Needles (right) data sets.

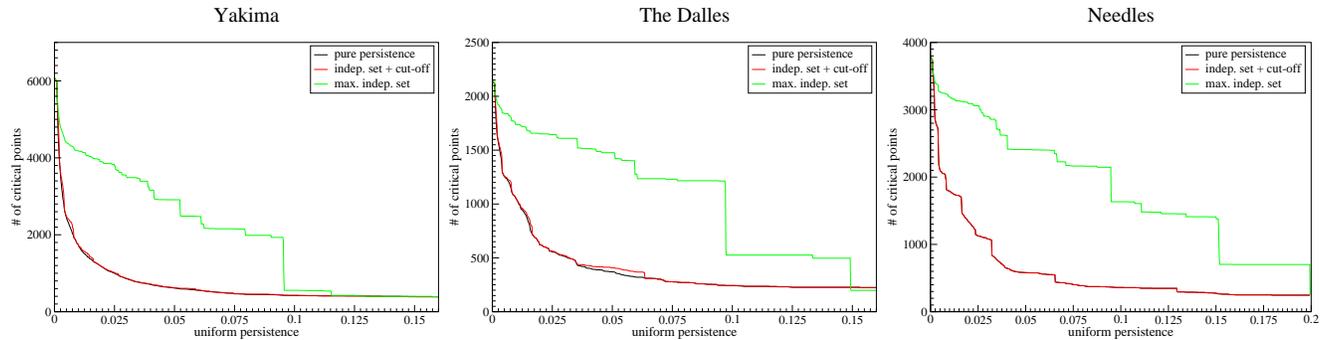


Fig. 17. Number of critical points in MS complex for the Yakima (left), The Dalles (middle), and the Needles (right) data sets.

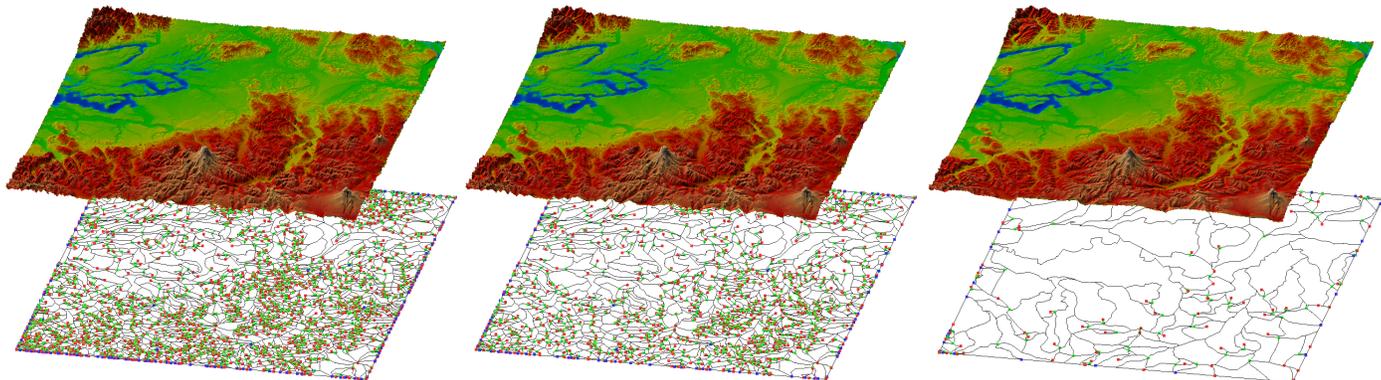


Fig. 18. (Left) Puget Sound data after topological noise removal. (Middle) Data at persistence of 1.2% of the maximum height. (Right) Data at persistence 20% of maximum height.

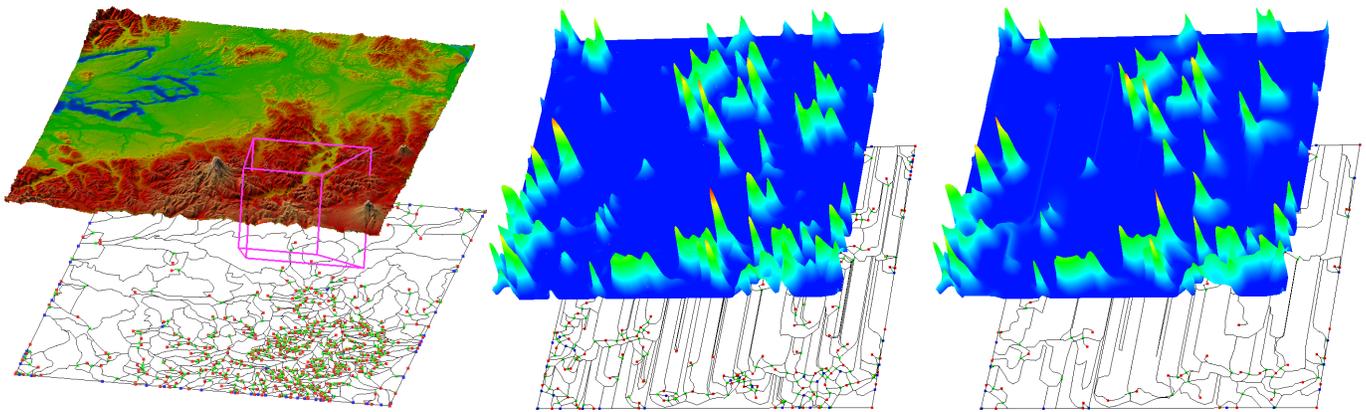


Fig. 19. (Left) View-dependent refinement of Puget Sound data (purple: view frustum). (Middle) Combustion data after topological noise removal. (Right) Adaptive refinement of combustion data based on function value. All maxima above 90% of the maximal function value are preserved.

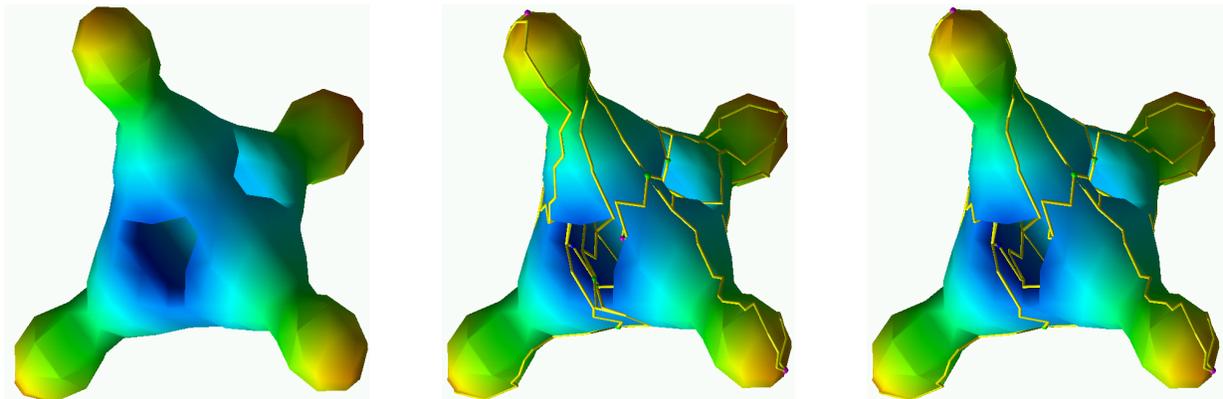


Fig. 20. Electrostatic and van der Waals potentials for a methane molecule. (Left) Isosurface of the electrostatic potential pseudo-colored with van der Waals potential. (Middle) Full MS complex with 30 critical points. (Right) Simplified MS complex with 14 critical points highlighting the hydrogen atoms near the maxima and the carbon atom near the minimum at the center.

VIII. CONCLUSIONS

We have described a new topology-based multi-resolution data structure for functions over a planar domain and demonstrated its use for two-dimensional height fields. The hierarchy allows one to extract geometry adaptively for a given topological error. Due to its robustness in the presence of noise and its well-defined simplification procedures, the approach is appealing for applications based on topological analysis, for example, data segmentation and feature detection and tracking in medical imaging or simulated flow field data sets. Future work will be concerned with fitting the complete geometry within a given error bound and the extension to volumetric data.

ACKNOWLEDGMENTS

This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48. Herbert Edelsbrunner is partially supported by the National Science Foundation (NSF) under grants EIA-99-72879 and CCR-00-86013. Bernd Hamann is supported by the NSF under contract ACI 9624034, through the LSSDSV program under contract ACI 9982251, and through the NPACI; the National Institute of Mental Health and the NSF under contract NIMH 2 P20 MH60975-06A2; the Lawrence Livermore National Laboratory under ASCI ASAP Level-2 Memorandum Agreement B347878 and under Memorandum Agreement B503159.

REFERENCES

- [1] A. Cayley, "On contour and slope lines," *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, vol. XVIII, pp. 264–268, 1859.
- [2] J. C. Maxwell, "On hills and dales," *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, vol. XL, pp. 421–427, 1870.
- [3] J. Pfaltz, "Surface networks," *Geographical Analysis*, vol. 8, pp. 77–93, 1976.
- [4] —, "A graph grammar that describes the set of two-dimensional surface networks," *Graph-Grammars and Their Application to Computer Science and Biology (Lecture Notes in Computer Science)*, vol. no. 73, 1979.
- [5] M. Morse, "Relations between the critical points of a real functions of n independent variables," *Transactions of the American Mathematical Society*, vol. 27, pp. 345–396, July 1925.
- [6] J. Milnor, *Morse Theory*. New Jersey: Princeton University Press, 1963.
- [7] H. Hoppe, "Progressive meshes," *Computer Graphics (Proc. SIGGRAPH '96)*, vol. 30, no. 4, pp. 99–108, Aug. 1996.
- [8] J. Popovic and H. Hoppe, "Progressive simplicial complexes," in *Proceedings of ACM SIGGRAPH 1997*, T. Whitted, Ed., vol. 31, ACM, New York, NY, USA: ACM Press / ACM SIGGRAPH, 1997, pp. 209–216. [Online]. Available: citeseer.nj.nec.com/article/popovic97progressive.html
- [9] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *Proceedings of ACM SIGGRAPH 1997*, T. Whitted, Ed., vol. 31, ACM, New York, NY, USA: ACM Press / ACM SIGGRAPH, 1997, pp. 209–216.
- [10] P. Lindstrom and G. Turk, "Fast and memory efficient polygonal simplification," in *Proc. IEEE Visualization '98*, IEEE, Los Alamitos California: IEEE Computer Society Press, Oct. 1998, pp. 279–286.
- [11] T. He, L. Hong, A. Varshney, and S. W. Wang, "Controlled topology simplification," *IEEE Trans. on Visualization and Computer*

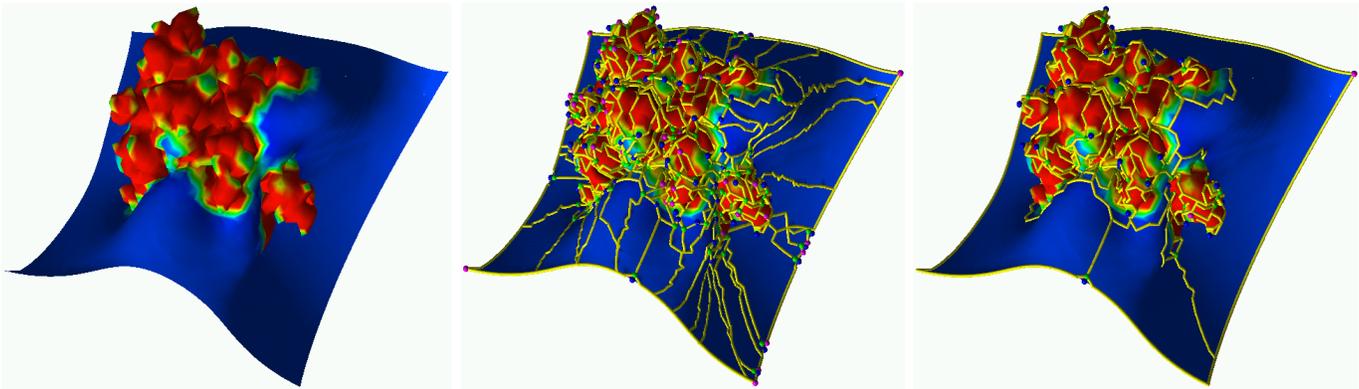


Fig. 21. Interaction energy between glucose and ethane under the three translational degrees of freedom. (Left) Isosurface of the electrostatic interaction pseudo-colored with the corresponding van der Waals potential. (Middle) Full MS complex with 564 critical points. (Right) Simplified MS complex with 166 critical points highlighting good candidate binding sites.

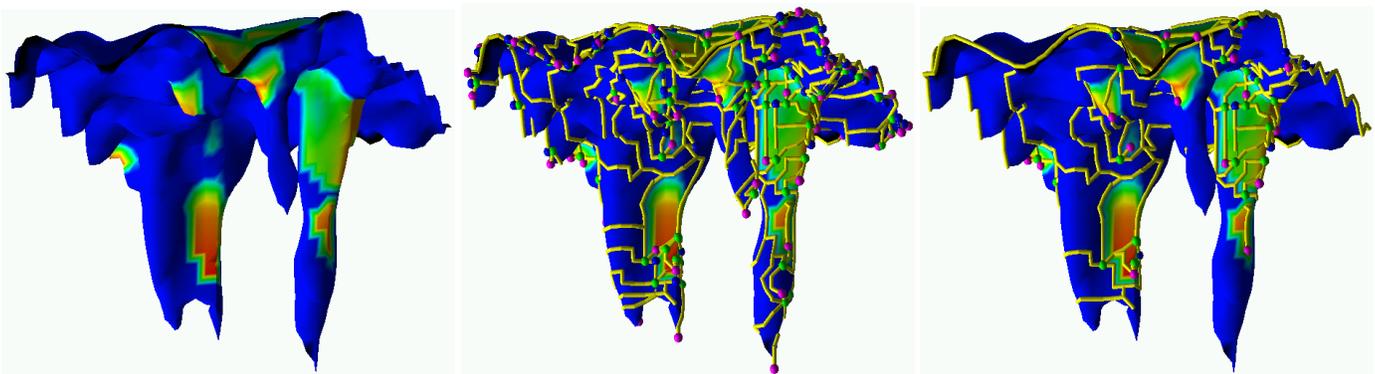


Fig. 22. Remediation process of contaminated ground. (Left) The domain M is an isosurface of the oil concentration in the soil (ground level on the top) showing the penetration of the oil spill. The pseudo-colored function f is the concentration of microbes consuming the oil. (Middle) Full MS complex of f with 232 critical points. (Right) Simplified MS complex with 67 critical points highlighting the main sites activity of the microbes.

- Graphics*, vol. 2, no. 2, pp. 171–184, 1996. [Online]. Available: citeseer.nj.nec.com/he96controlled.html
- [12] J. El-Sana and A. Varshney, “Topology simplification for polygonal virtual environments,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 2, pp. 133–144, 1998. [Online]. Available: citeseer.nj.nec.com/el-sana98topology.html
- [13] J. L. Helman and L. Hesselink, “Visualizing vector field topology in fluid flows,” *IEEE Computer Graphics and Applications*, vol. 11, no. 3, pp. 36–46, May/June 1991.
- [14] W. de Leeuw and R. van Liere, “Collapsing flow topology using area metrics,” in *Proc. IEEE Visualization '99*. IEEE Computer Society Press, 1999, pp. 349–354.
- [15] X. Tricoche, G. Scheuermann, and H. Hagen, “A topology simplification method for 2d vector fields,” in *Proc. IEEE Visualization '00*, IEEE. Los Alamitos California: IEEE Computer Society Press, 2000, pp. 359–366.
- [16] —, “Continuous topology simplification of planar vector fields,” in *Proc. IEEE Visualization '01*, IEEE. Piscataway, NJ: IEEE Computer Society Press, Oct. 2001, pp. 159–166.
- [17] A. T. Fomenko and T. L. Kunii, Eds., *Topological Modeling for Visualization*. Springer-Verlag, 1997.
- [18] C. L. Bajaj and D. R. Schikore, “Topology preserving data simplification with error bounds,” *Computers and Graphics*, vol. 22, no. 1, pp. 3–12, 1998.
- [19] K. Hormann, “Morphometrie der Erdoberfläche, Schrift. d. Univ. Kiel,” 1971.
- [20] D. M. Mark, “Topological properties of geographic surfaces,” in *Proc. Advanced Study Symposium on Topological Data Structures*, Oct. 1977.
- [21] H. Edelsbrunner, D. Letscher, and A. Zomorodian, “Topological persistence and simplification,” *Discrete Comput. Geom.*, vol. 28, pp. 511–533, 2002.
- [22] H. Edelsbrunner, J. Harer, and A. Zomorodian, “Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds,” *Discrete Comput. Geom.*, vol. 30, pp. 87–107, 2003.
- [23] H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci, “Morse-Smale complexes for piecewise linear 3-manifolds,” in *Proc. 19th Sympos. Comput. Geom.*, 2003, pp. 361–370.
- [24] Y. Matsumoto, *An Introduction to Morse Theory*. American Mathematical Society, 2002.
- [25] J. R. Munkres, *Elements of Algebraic Topology*. Redwood City, CA: Addison-Wesley, 1984.
- [26] P. S. A. v, *Combinatorial Topology*. New York: Dover, 1998.
- [27] H. Edelsbrunner and E. P. Mücke, “Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms,” *ACM Trans. Graphics (TOG)*, vol. 9, pp. 66–104, 1990.
- [28] R. Carlson and F. N. Fritsch, “Monotone piecewise bicubic interpolation,” *Journal on Numerical Analysis*, vol. 22, pp. 386–400, Apr. 1985.
- [29] H. Greiner, “A survey on univariate data interpolation and approximation by splines of given shape,” *Math. Comput. Model.*, vol. 15, pp. 97–106, 1991.
- [30] G. Taubin, “A signal processing approach to fair surface design,” *Computer Graphics (Proc. SIGGRAPH '95)*, pp. 351–358, 1995.
- [31] M. S. Floater, “Mean value coordinates,” *Computer Aided Geometric Design*, vol. 20, pp. 19–27, 2003.
- [32] A. Balázs, M. Guthe, and R. Klein, “Fat borders: Gap filling for efficient view-dependent lod rendering,” Universität Bonn, Tech. Rep. CG-2003-2, June 2003.
- [33] T. W. Sederberg, J. Zheng, A. Bakenov, and A. Nasri, “T-splines and t-nurcs,” *ACM Transactions on Graphics (TOG)*, vol. 22, no. 3, pp. 477–484, 2003.
- [34] P. Linstrom and V. Pascucci, “Terrain simplification simplified: A general framework for view-dependent out-of-core visualization,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 8, no. 3, pp. 239–254, July–September 2002.
- [35] T. Echehki and J. H. Chen, “Direct numerical simulation of autoignition in non-homogeneous hydrogen-air mixtures,” *Combust. Flame*, 2003, to appear.