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 a function over a two-dimensional domain. 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. Examples of such data are 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 error measurement. 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 the 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.

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].

*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

In the field of flow visualization, topological analysis and topology based simplification originates with the work of Helman and Hesselink [13]. They show how to find and classify critical points in flow fields and proposed a structure similar to the Morse-Smale complex to analyze vector fields. Later methods to simplify this complex based on different error bounds 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-Smale complexes for data sets with tens of thousands of critical points compared to hundreds of critical points in commonly used vector fields. Unlike the method in [14] we maintain a consistent geometric approximation of the topology. Furthermore, we avoid creating 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 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 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 into 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 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 functions 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 geometric methods are used to realize monotonic paths and patches. The latter play 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 $f : \mathbb{M} \to \mathbb{R}$ 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 a. 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 one 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 overlayed complexes, which are the minima and maxima of f, together with the crossing points of ascending and descending 1-manifolds, which are the saddles. 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 the center of Fig. 1.

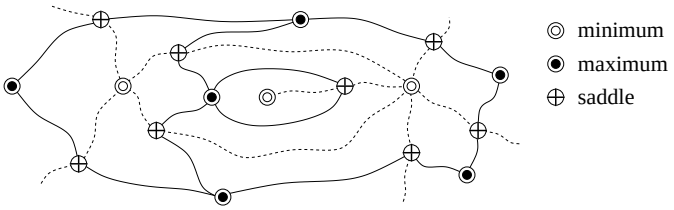


Fig. 1. The folded quadrangle in the middle of this MS complex has two boundary arcs glued to each other.

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. Following [27], 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. 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.

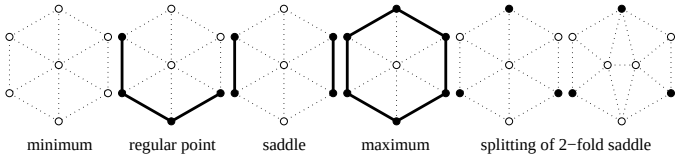


Fig. 2. Classification of a vertex based on relative height of vertices in its link. The lower link is marked black.

D. Persistence

We require a numerical measure of the importance of critical points that can be used to drive the simplification of an 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. Note that once two paths have merged they never separate.

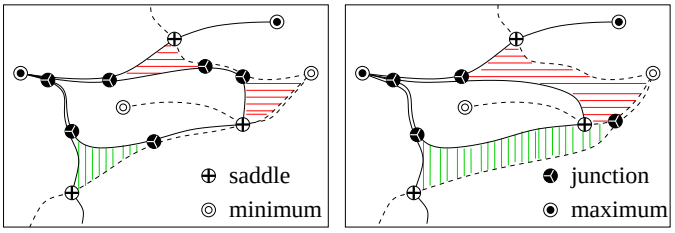


Fig. 3. Portion of the MS complex of a piecewise linear function. Since the gradient is not continuous, ascending (solid) and descending (dotted) paths can meet in junctions and share segments. Left: complex with no restrictions on sharing segments. The green region touches only one saddle, and the red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

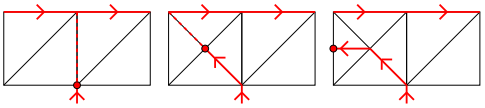
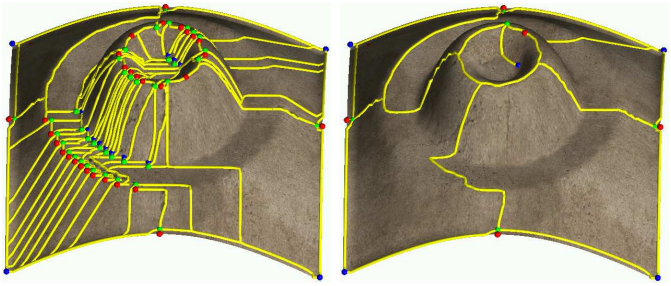


Fig. 4. Triangle split to keep two paths separated. Solid red lines indicate two portions of paths already computed. Left: the red circle is the current extremum of a path that would follow the red dotted line. Middle: the path is extended splitting a first triangle. Right: since the two paths would 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) [28]. 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. This simulates a set of arbitrarily small perturbations resolving all degeneracies. 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.



(a) (b)

Fig. 5. MS complex of degenerate data set. The "volcano" is created by the rotational sweep of a 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 removing symbolic features with zero persistence.

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

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;
$S =$ FINDSADDLES(T);
$S =$ SPLITMULTIPLESADDLES(T);
SORTBYHEIGHT(S);
`forall` $s \in S$ in ascending order `do`
  COMPUTEASCENDINGPATH(P)
`endfor`;
`forall` $s \in S$ in descending order `do`
  COMPUTEDESCENDINGPATH(P)
`endfor`;
`while` there exists untouched $f \in F$ `do`
  GROWREGION(f, p_0, p_1, p_2, p_3);
  CREATEMORSECELL(C, p_0, p_1, p_2, p_3)
`endwhile`;
$M =$ 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 four boundary paths, p_0 to p_3, which we incorporated into a half-edge representation of the cell.

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.

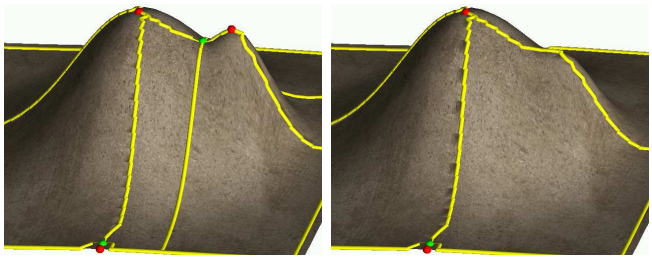


Fig. 7. Portion of the graph of a function before (left) and after (right) cancellation of a maximum (red) and a saddle (green).

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 recomputed 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 by showing each diamond

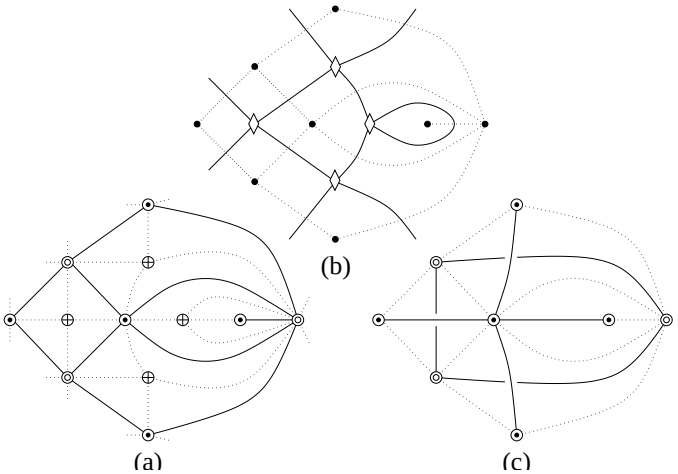


Fig. 8. (a) The portion of an MS complex (dotted) and the portion of the corresponding decomposition into diamonds (solid). (b) Portion of the data structure (solid) representing the piece of the decomposition into diamonds (dotted). (c) Cancellation graph (solid) of the decomposition into diamonds (dotted).

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.

To construct the hierarchy by repeated cancellations, we use the algorithm in [21] to match critical points

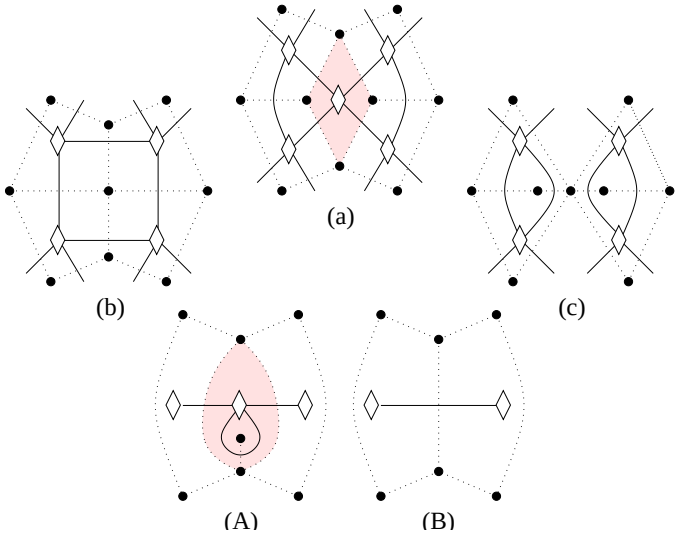


Fig. 9. The four-sided diamond (a) can be zipped up in two ways: from top to bottom (b) or from left to right (c). A folded diamond (A) can be zipped up in only one way (B).

in pairs $(s_1, v_1), (s_2, v_2), \ldots, (s_k, v_k)$, 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*. This notion of dependencies is similar to methods used is view-dependent refinements of polygon meshes [29], [30] applied to the MS complex. 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.

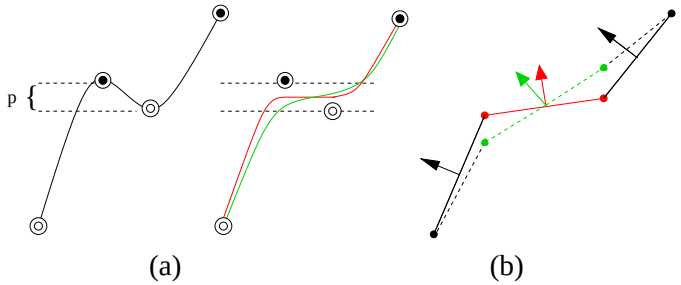


Fig. 10. Geometry fitting for paths: (a) One-dimensional cancellation and several monotonic approximations. (b) Local averaging used to construct smoothly varying monotonic approximations. Slopes of neighboring edges are combined with the original slope, and the function values are adjusted accordingly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

B. Data fitting

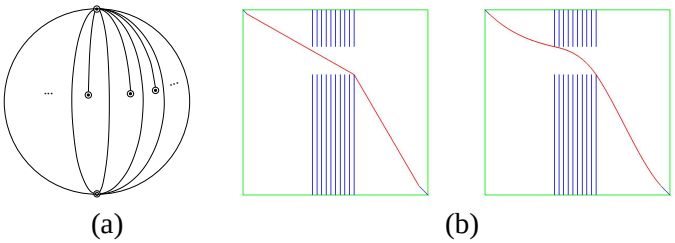


Fig. 11. (a) MS complex on the sphere with pairwise dependent cancellations. (b) One-dimensional gradient smoothing with (blue) error constraints and prescribed endpoint derivatives. Left: initial configuration. Right: constructed solution.

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 [31], [32]. 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 therefore did not adapt standard techniques for our purposes and instead 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 [33]. 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 paths constructed in Step 4 are not guaranteed to satisfy the claimed error bounds, which is the reason for the repeated use of gradient smoothing in Step 5. We iterate until the paths are monotone and satisfy the error bounds. Experimentally, it takes only a constant number of iterations to achieve both goals. In our software, the fitting of the surface is currently the bottleneck due to slow convergence iteratively solving a Laplacian system.

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. For maximal flexibility, the triangulation for each region should not depend on neighboring regions.

A. Path smoothing

Without modifications the algorithms used to compute paths tend to create jagged paths on the 2-manifold, as in Fig. 12(a). These are visually not pleasing and difficult to approximate. We therefore slightly modify the data to obtain smoother paths, again using Laplacian smoothing. Special care has to be taken at junctions, where we separately average the predecessor and the successor vertices before updating 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 the layout of the paths.

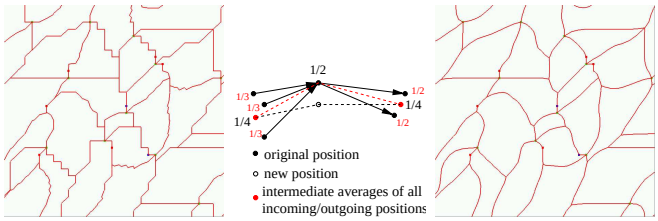


Fig. 12. Path smoothing: (a) A typical path structure without smoothing. (b) Smoothing applied at junctions. (c) The path structure of (a) after two smoothing steps.

B. Parametrization

To enable fast and versatile rendering of the data and reduce memory requirements, we remesh each quadrangular region using a regular grid structure. First, we compute a mapping of the boundary of the region to the boundary of one or more unit squares, see below. Then we use standard parametrization techniques such as [34], [35] to extend the mapping to the interior. 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 boundary parametrization is chosen such that the meshes of neighboring regions agree geometrically along their boundary.

C. 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 always contains its critical points and junctions. The critical points represent the extremal function values of a region. 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. As base-shape in parameter space we use one or more unit squares, and we choose the number depending on the ratio of the eigenvalues of the principle component analysis of all boundary vertices. Once the base-shape is known, the critical points and junctions are fitted recursively using arc-length parametrization. The complete process is illustrated in Figure 13.

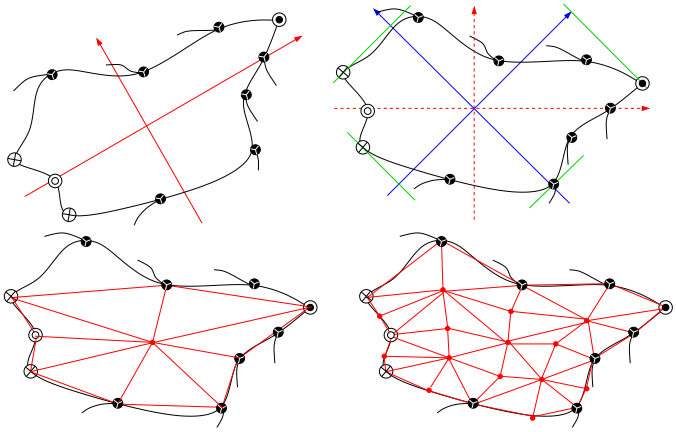


Fig. 13. Creating a parametrization for the boundary. Top-left: original region and local coordinate system defined by the principal component analysis. Top-right: region after transformation into the new coordinate system. We use a single unit square as base-shape in parameter space. The points with extreme projections onto the two diagonals are mapped to the corners of the base-shape. Bottom-left: regular mesh after the first level of recursively fitting the junctions. Bottom-right: Final mesh.

To remesh the path segments between critical points and junctions we apply midpoint subdivision based on arclength. 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. [36].

TABLE I
DATA SETS USED FOR TESTING. THE FIRST THREE ARE TERRAINS.

	resolution	function values
Puget Sound	1025×1025	2 byte int
Needles	1201×1201	1 byte int
Dalles	1201×1201	1 byte int
Glucose-Ethane	5215 vertices	fbat
Oil Spill	3231 vertices	fbat

VII. RESULTS

We have applied our algorithm to terrain data converted from digital elevation models[1] and to isosurfaces from two scientific data sets, all listed in Table I. The Glucose-Ethane data set in Fig. 16 describes the interaction energy between a ligand (glucose) and a receptor (ethane) under the three translational degrees of freedom. The domain is an isosurface of the electrostatic potential and the function is the van der Waals energy. The Oil Spill data set in Fig. 17 shows a ground remediation process after an oil spill contamination. The domain is an isosurface of the oil concentration, reaching from the ground level at the top down into the soil. The superimposed pseudo-colored function shows the concentration of microbes consuming the oil and performing the remediation process. In both cases the hierarchical MS complex highlights regions of interest such as good candidate bonding sites for molecular interaction or regions of high microbe activity in the ground 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 step should always be applied, even if only to remove the artifacts caused by symbolic perturbation. We classify all features with persistence below 0.1% of the total function range as noise. Fig. 14 illustrate 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 clean-up is a valuable pre-processing step for many techniques proposed in recent years.

We have tested three strategies for creating the hierarchy: sequential, batched, and hybrid. The sequential strategy performs the cancellations in the order of increasing persistence. The batched strategy removes a maximal independent set of cancellations in one step, collecting a batch greedily in the order of increasing persistence. Finally, the hybrid strategy is the batched

[1] http://www.webgis.com

TABLE II
STATISTICS ON THE HIERARCHIES FOR COMPARING THE THREE CANCELLATION STRATEGIES.

	depth	avg dep	max #p	max #c	avg deg
Puget Sound	49,185 original vs. 17.470 significant critical points				
sequential	381	128	148	110	3.80
batched	157	118	131	112	4.28
hybrid	238	105	147	106	3.94
Dalles	24,617 original vs. 2,144 significant critical points				
sequential	80	34	75	39	3.40
batched	54	31	82	43	3.88
hybrid	73	33	63	57	3.52
Needles	17,375 original vs. 3,772 significant critical points				
sequential	177	68	111	87	3.60
batched	113	70	87	87	3.88
hybrid	149	62	124	101	3.68

strategy with the added restriction that the largest persistence be at most twice the smallest persistence in the same batch. We limit each strategy to critical points whose persistence does not exceed 20% of the function range.

Table II summarizes the collected statistics for the hierarchies constructed from the three terrain data. For each combination of data set and cancellation strategy, it lists the maximum and average depth of the leaves, the maximum number of parents and children of the nodes, and the average degree, defined as the combined number of parents and children of a node. As expected, the batched cancellation of critical points creates more shallow hierarchies than the sequential strategy. However, this is not always the case. In the Needles data, the average depth created by batching exceeds that created by sequential cancellation. This observation can be explained by the existence of high-degree nodes illustrated in Fig. 14 which shows the highest-resolution MS complex of the Needles data, drawing each path as a straight line from the saddle to the extremum. There are very few minima in the data forcing a large average degree and an uneven distribution of nodes over the levels of the hierarchy.

The graphs in Fig. 15 show the number of nodes per level for the Puget Sound and Dalles data set. The batched strategies clearly produce superior results in terms of overall shape of the hierarchy. However, this does not necessarily translate into better performance in practice. Fig. 15 also shows the number of critical points in the MS complex depending on a uniform error. Even though the hierarchy created by batched cancellation is the most shallow it also contains significantly denser meshes. The hybrid strategy combines the advantages of the other two strategies and is therefore our method of choice.

VIII. CONCLUSIONS

We have described a new topology-based multi-resolution data structure for real-valued functions over two-dimensional domains and demonstrated its use for terrains. The hierarchy allows for the adaptive extraction of geometry depending on given topological error. Due to its robustness in the presence of noise and its well-defined simplification procedures, the approach is appealing for applications that rely on topological analysis. Examples are 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 (NFS) 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," *London, Edinburgh and Dublin Phil. Mag. J. Sci.*, vol. XVIII, pp. 264–268, 1859.
[2] J. C. Maxwell, "On hills and dales," *London, Edinburgh and Dublin Phil. Mag. J. Sci.*, 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*. Springer-Verlag, Lecture Notes in Computer Science, vol. 73, 1979.
[5] M. Morse, "Relations between the critical points of a real function of *n* independent variables," *Trans. Amer. Math. Soc.*, vol. 27, pp. 345–396, 1925.
[6] J. Milnor, *Morse Theory*. New Jersey: Princeton Univ. Press, 1963.
[7] H. Hoppe, "Progressive meshes," *Comput. Graphics (Proc. SIGGRAPH)*, vol. 30, pp. 99–108, 1996.
[8] J. Popovic and H. Hoppe, "Progressive simplicial complexes," *Computer Graphics (Proc. SIGGRPAH)*, vol. 31, pp. 209–216, 1997.
[9] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," *Comput. Graphics (Proc. SIGGRAPH)*, vol. 31, pp. 209–216, 1997.
[10] P. Lindstrom and G. Turk, "Fast and memory efficient polygonal simplification," in *Proc. IEEE Visualization*, 1998, pp. 279–286.
[11] T. He, L. Hong, A. Varshney, and S. W. Wang, "Controlled topology simplification," *IEEE Trans. Visual. Comput. Graphics*, vol. 2, pp. 171–184, 1996.

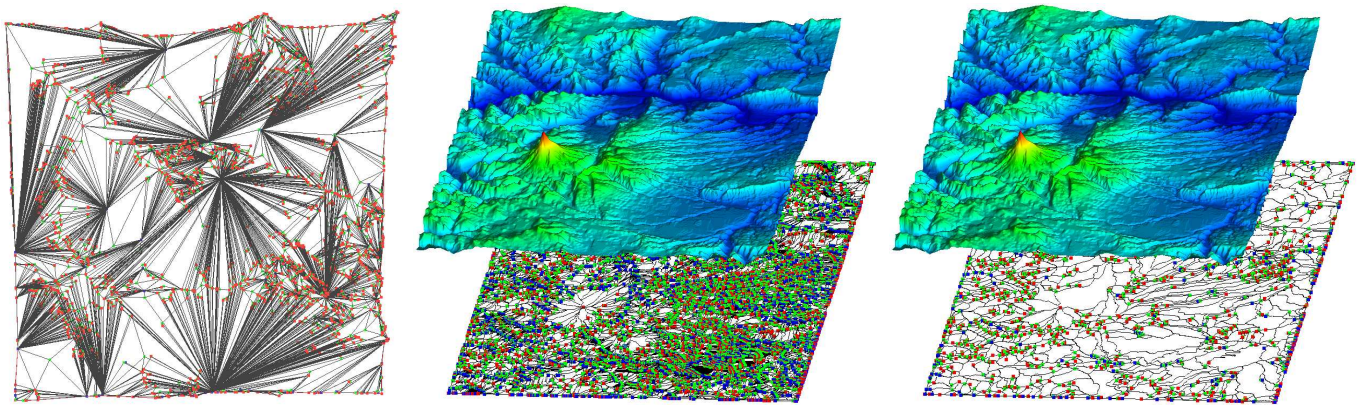


Fig. 14. Left: highest resolution MS complex of Needles data set. Middle: original Dalles data set containing 24,617 critical points. Right: same data with 2,144 critical points after removing all with persistence less than 0.1% of height range.

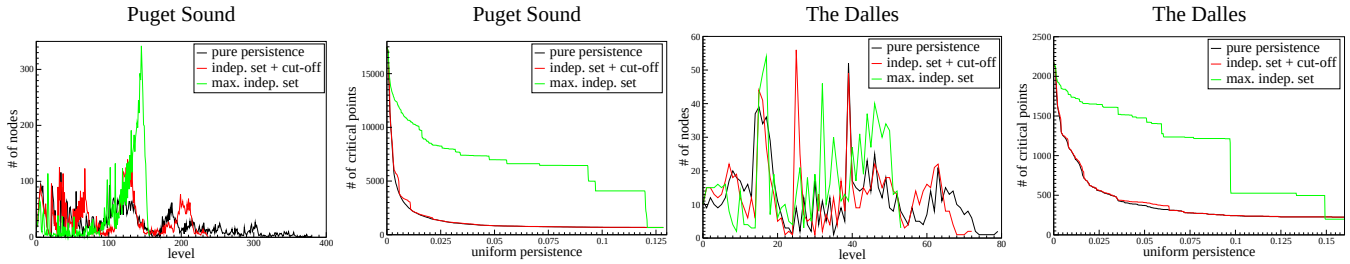


Fig. 15. Nodes distribution over the levels for three cancellation strategies and number of critical points in MS complex for the Puget Sound and the Dalles data sets.

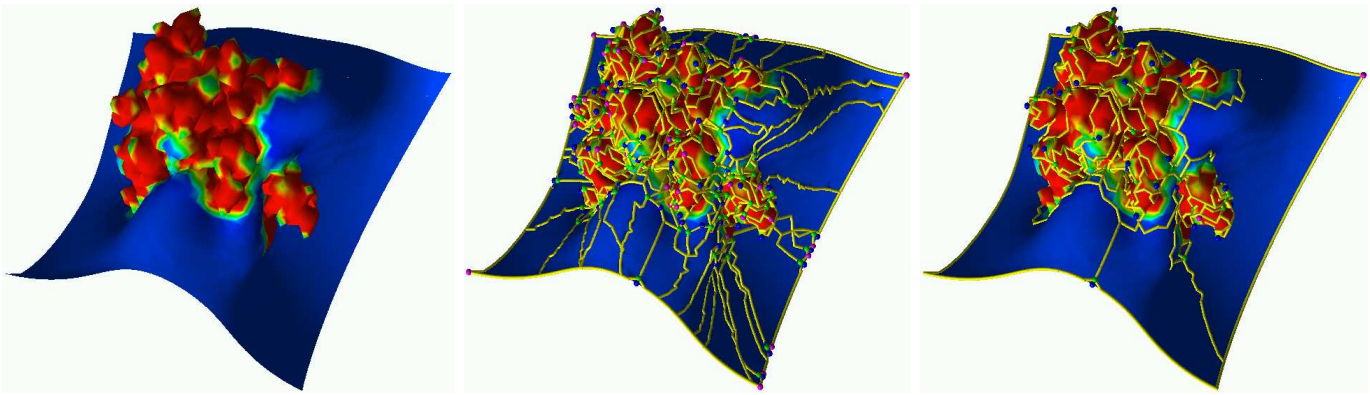


Fig. 16. Interaction energy between glucose and ethane under the three translational degrees of freedom. Left: isosurface of the electrostatic interaction pseudo-colored with the corresponding van der Waals potential. Middle: full MS complex with 564 critical points. Right: simplified MS complex with 166 critical points highlighting good candidate binding sites.

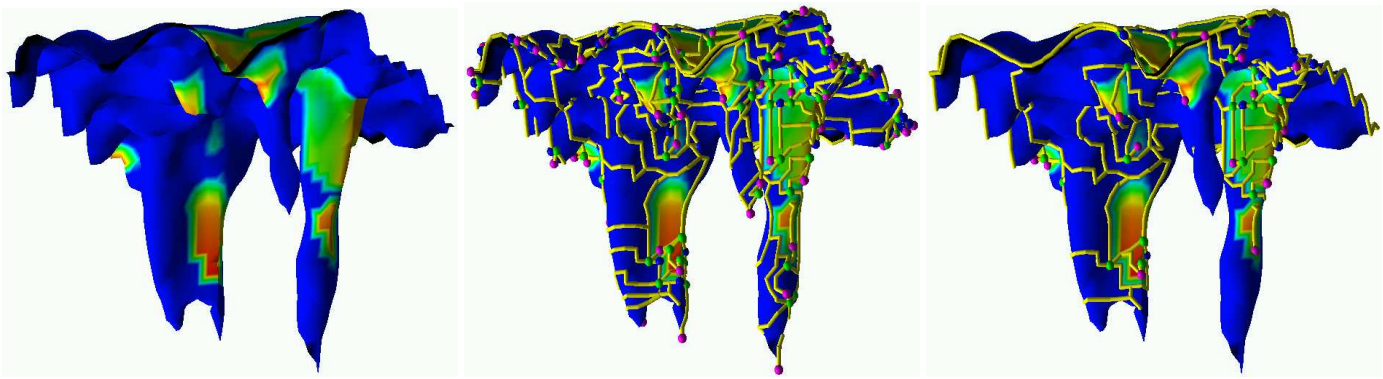


Fig. 17. Remediation process of contaminated ground. Left: the isosurface of the oil concentration in soil with a function in pseudo-color measuring the density of microbes consuming the oil. Middle: full MS complex with 232 critical points. Right: simplified MS complex with 67 critical points highlighting the main activity sites of the microbes.

[12] J. El-Sana and A. Varshney, "Topology simplification for polygonal virtual environments," *IEEE Trans. Visual. Comput. Graphics*, vol. 4, pp. 133–144, 1998.

[13] J. L. Helman and L. Hesselink, "Visualizing vector field topology in fluid flows," *IEEE Comput. Graphics Appl.*, vol. 11, pp. 36–46, 1991.

[14] W. de Leeuw and R. van Liere, "Collapsing flow topology using area metrics," in *Proc. IEEE Visualization*, 1999, pp. 349–354.

[15] X. Tricoche, G. Scheuermann, and H. Hagen, "A topology simplification method for 2d vector fields," in *Proc. IEEE Visualization*, 2000, pp. 359–366.

[16] ——, "Continuous topology simplification of planar vector fields," in *Proc. IEEE Visualization*, 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, pp. 3–12, 1998.

[19] K. Hormann, "Morphometrie der Erdoberfläche," *Schrift. Univ. Kiel*, 1971.

[20] D. M. Mark, "Topological properties of geographic surfaces," in *Proc. Adv. Study Sympos. Topol. Data Structures*, 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 Ann. Sympos. Comput. Geom.*, 2003, pp. 361–370.

[24] Y. Matsumoto, *An Introduction to Morse Theory*. Amer. Math. Soc., 2002.

[25] J. R. Munkres, *Elements of Algebraic Topology*. Redwood City, CA: Addison-Wesley, 1984.

[26] P. S. Alexandrov, *Combinatorial Topology*. New York: Dover, 1998.

[27] T. F. Banchoff, "Critical points for embedded polyhedral surfaces," *Amer. Math. Monthly*, vol. 77, pp. 457–485, 1970.

[28] H. Edelsbrunner and E. P. Mücke, "Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms," *ACM Trans. Graphics*, vol. 9, pp. 66–104, 1990.

[29] J. C. Xia and A. Varshney, "Dynamic view-dependent simplification for polygonal models," in *Proc. IEEE Visualization*, 1996, pp. 335–344.

[30] H. Hoppe, "View-dependent refinement of progressive meshes," *Comput. Graphics (Proc. SIGGRAPH)*, vol. 31, pp. 189–98, 1997.

[31] R. Carlson and F. N. Fritsch, "Monotone piecewise bicubic interpolation," *J Num. Anal.*, vol. 22, pp. 386–400, 1985.

[32] H. Greiner, "A survey on univariate data interpolation and approximation by splines of given shape," *Math. Comput. Model.*, vol. 15, pp. 97–106, 1991.

[33] G. Taubin, "A signal processing approach to fair surface design," *Computer Graphics (Proc. SIGGRAPH)*, pp. 351–358, 1995.

[34] M. S. Floater, "Mean value coordinates," *Comput. Aided Geom. Design*, vol. 20, pp. 19–27, 2003.

[35] M. Desbrun, M. Meyer, and P. Alliez, "Intrinsic parameterizations of surface meshes," *Comput. Graphics Forum (Proc. Eurographics)*, vol. 21, 2002.

[36] A. Balázs, M. Guthe, and R. Klein, "Fat borders: gap filling for efficient view-dependent lod rendering," Univ. Bonn, Germany, Tech. Rep. CG-2003-2, 2003.

Peer-Timo Bremer is currently pursung his Ph.D. in computer science at the University of California at Davis. He holds a student employee graduate research fellowship at the Lawrence Livermore National Laboratory. He obtained a Diplom (M.S.) in mathematics with a second major in computer science from the University of Hannover, Germany, in 2000. He is a member of the Association for Computing Machinery (ACM) and the Institute of Electrical and Electronics Engineers (IEEE) Computer Society.

Herbert Edelsbrunner is currently Arts and Sciences Professor of Computer Science and Mathematics at Duke University. He is also adjunct professor at the University of North Carolina at Chapel Hill and director at Raindrop Geomagic, a reverse engineering and geometric modeling company he co-founded with Ping Fu in 1996. He holds Dipl.-Ing. and Ph.D. degrees from the Graz University of Technology in Austria. Herbert Edelsbrunner published two books in the general area of geometric algorithms, the first in 1987 on "Alogrithms in combinatorial geometry" with Springer-Verlag and the second in 2001 on "Geometry and topology for mesh generation" with Cambridge University Press. He received the Alan T. Waterman Award from the National Science Foundation in 1991.

Bernd Hamann serves as associate vice chancellor for research, is co-director of the Center for Image Processing and Integrated Computing (CIPIC), and full professor of computer science at the University of California, Davis. Bernd Hamann received a B.S. in computer science, a B.S. in mathematics, and an M.S. in computer science from the Technical University of Braunschweig, Germany. He received a Ph.D. in computer science from Arizona State University in 1991. He was awarded a 1992 Research Initiation Award by Mississippi State University, a 1992 Research Initiation Award by the National Science Foundation, and a 1996 CAREER Award by the National Science Foundation. In 1995, he received a Hearin-Hess Distinguished Professorship in Engineering by the College of Engineering at Mississippi State University. Bernd Hamann is a member of the Association for Computing Machinery, the Institute of Electrical and Electronics Engineers, the Society for Industrial and Applied Mathematics, and the IEEE Technical Committee on Visualization and Graphics.

Valerio Pascucci is a Computer Scientist and Project Leader at Lawrence Livermore National Laboratory, Center for Applied Scientific Computing since May 2000. Prior to his CASC tenure, he was a senior research associate at the University of Texas at Austin, Center for Computational Visualization, CS and TICAM Departments. Valerio earned a Ph.D. in computer science at Purdue University in May 2000, and a EE Laurea (Master), at the University "La Sapienza" in Roma, Italy, in December 1993, as a member of the Geometric Computing Group. Valerio came to the U.S. in 1995 after having grown up in Roma, Italy.