Maximizing Adaptivity in Hierarchical Topological Models

Peer-Timo Bremer  
Dept. of Computer Science,  
University of Illinois,  
Urbana-Champaign  
ptbremer@acm.org

Valerio Pascucci  
Center for Applied Scientific Computing  
Lawrence Livermore National Laboratory  
pascucci@llnl.gov

Bernd Hamann  
Institute for Data Analysis and Visualization  
Dept. of Computer Science  
University of California, Davis  
hamann@cs.ucdavis.edu

ABSTRACT

We present an approach to hierarchically encode the topology of functions over triangulated surfaces. We describe the topology of a function by its Morse-Smale complex, a well known structure in computational topology. Following concepts of Morse theory, a Morse-Smale complex (and therefore a function’s topology) can be simplified by successively canceling pairs of critical points. We demonstrate how cancellations can be effectively encoded to produce a highly adaptive topology-based multi-resolution representation of a given function. Contrary to the approach of [4] we avoid encoding the complete complex in a traditional mesh hierarchy. Instead, we encode a reduced complex created by disregarding some topological constraints on the complex. The corresponding data is stored separately in a structure called cancellation forest. Conceptually, a cancellation forest consists of sets of critical points governed by the concepts of Morse theory. The combination of this new structure with a traditional mesh hierarchy proofs to be significantly more flexible than the one previously reported [4]. In particular, the resulting hierarchy is guaranteed to be of logarithmic height.

1. INTRODUCTION

Topology-based methods used for visualization and analysis of scientific data are becoming increasingly popular. Their main advantage lies in the capability to provide a concise description of the overall structure of a scientific data set. Subtle features can easily be missed when using “traditional” visualization methods like volume rendering or contouring, unless “correct” transfer functions and isovales are chosen. On the other hand, the presence of a large number of small features creates a “noisy visualization,” in which larger features can be overlooked. By visualizing topology directly, one can guarantee that no feature is missed. Furthermore, one can use sound mathematical principles to simplify a topological structure. The topology of functions is also often used for feature detection and segmentation (e.g., in surface segmentation based on curvature).

However, for topology-based data analysis one needs flexible, hierarchical models able to adaptively remove noise or features not relevant for a particular segmentation. In practice, the simplification/refinement should be fast (possibly interactive) and highly adaptive in order to be useful for a large variety of situations. Requiring interactivity inadvertently leads to the use of hierarchical encodings rather than simplification schemes. Hierarchical models often reduce the adaptivity of a representation to gain the ability to perform incremental changes for varying queries.

We address the need for adaptive topology-based data exploration by improving significantly the topological hierarchy described in [4]. By encoding a less specific complex as multi-resolution mesh and the remaining information separately as a collection of sets, we show how one can remove many of the dependencies in the original hierarchy, making the structure simpler, more compact, and more adaptive than the original one.

1.1 Related Work

The topological structure of a scalar field can be described partially by its contour tree [21, 5, 22], which describes the relations between the connected components of its level sets. This structure provides a user with a compact representation of the topology [1] and can be used to accelerate the computation of isosurfaces [30]. However, the contour tree provides only limited information about the embedding of the level sets and therefore remains somewhat abstract. Morse theory [20, 19], on the other hand, provides methods to analyze the complete topology of a function over a manifold as well as its embedding. Early approaches for the bivari- case are discussed in [6, 18, 23]. More recently, the Morse-Smale complex was introduced by Edelsbrunner et al. [10, 9] as a description of the topology of scalar-valued functions over two- and three-dimensional manifolds. Applications of this theory vary from implicit geometry modeling [27] to shape description [15]. Related concepts are also used in flow visualization. Helman and Hesselink [14] showed how to find and classify critical points in flow fields and proposed a structure similar to the Morse-Smale complex for vector fields. Later, methods to analyze and simplify this complex were proposed by de Leeuw and van Liere [8] and Tricoche et al. [28, 29].

A large amount of research has been devoted to the construction of multi-resolution representations for meshes. Even a partial overview of the related work is beyond the scope of this paper. Conceptually, the hierarchy for the reduced Morse-Smale complex is related to concepts from [25, 13, 26, 16, 17], as it is created by successive vertex removal. For more references and a discussion on a general theory of multi-resolution modeling we refer the reader to [12]. Nevertheless, the complete Morse-Smale complex incorporates many topological constraints, and its structure (and multi-resolution encoding) is quite different from those of traditional meshes. The first multi-resolution encoding of a Morse-Smale complex we are aware of was proposed by Pfaltz [24], which has been improved and extended by Edelsbrunner et al. [10] and Bremer et al. [3, 4]. The more recent hierarchical structures are based on the concept of persistence [11], which relates the difference in function value of critical point pairs to the importance of a
Given a Morse-Smale complex, we

1. introduce the concept of a reduced Morse-Smale complex;
2. define a cancellation forest to encode the difference between a reduced and complete complex;
3. describe a multi-resolution encoding for both structures, including quality guarantees; and
4. apply our algorithms to a variety of data sets.

We first review necessary concepts from Morse theory and the construction of a Morse-Smale complex (Section 2). In Section 3, we describe the cancellation forest, and we discuss the resulting hierarchy in Section 4. We conclude with results and possibilities for future research (Section 5).

2. MORSE-SMALE COMPLEX

We base our algorithms on intuitions derived from the study of smooth functions. We review key aspects from Morse theory [20, 19] for smooth functions and discuss how these can be used in the piecewise linear case.

2.1 Morse Theory

Given a smooth function \( f : \mathcal{M} \rightarrow \mathbb{R} \), a point \( a \in \mathcal{M} \) is called critical when its gradient \( \nabla f(a) = (\partial f / \partial x, \partial f / \partial y) \) vanishes; it is called regular otherwise. For two-manifolds, (non-degenerate) critical points are maxima (\( f \) decreases in all directions), minima (\( f \) increases in all directions), or saddles (\( f \) switches between decreasing and increasing four times around the point). Using a local coordinate frame at \( a \), we compute the Hessian \( H \) of \( f \), which is the matrix of second partial derivatives. If \( H \) is non-singular we can 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 \( a \). 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.

At any regular point, the gradient (vector) is non-zero, and when we follow the gradient 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 \( f \) is smooth, two integral lines are either disjoint or the same. 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 all integral lines that end at \( a \). The collection of descending 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 \( a \). When neglecting certain degenerate cases, see [10], we can overlay these two complexes and obtain what we call the Morse-Smale complex, or MS complex, of \( f \). Its nodes are the vertices of the two overlayed complexes, which are the minima, maxima, and saddles of \( f \). Its arcs are integral lines between saddles and extrema that bound four-sided regions. An example is shown in Figure 1.

Using the insight gained from smooth Morse theory when applied to piecewise linear functions, we follow the concepts described in [3]. We identify and classify critical points based on their local neighborhood, see [2, 10]. If all vertices that are edge-connected to a point \( v \) have function values below that of \( v \), we call it a maximum; if all are above \( v \), then we call it a minimum etc., see Figure 2. In general, there can exist saddles with multiplicity greater than one that we split into simple ones, as shown on the far right in Figure 2.

2.2 Variance

As a numerical measure of the importance of critical points we define pairs of critical points and use the absolute difference between their height/function values. Given a two-manifold \( \mathcal{M} \), one can imagine sweeping a plane through \( \mathcal{M} \) in the direction of increasing height (with respect to the scalar field value.) The topology of the part of \( \mathcal{M} \) below the sweep plane changes whenever we add a critical vertex, and it remains unchanged whenever we add a regular vertex. Except for some special cases, each change either creates a component or it destroys a component. In general, the sweeping order does not need to be tied to the function. In [11], Edelsbrunner et al. used a filtration of a simplicial complex as sweeping “plane” to define a unique pairing between critical points; a vertex \( v \) that creates a component is paired with the vertex \( u \) that destroys this component. The persistence of the pair \( u,v \) is defined as the “delay” between the two events.

We are mostly interested in the topology of the function and thus use a sweep by function value. Furthermore, the unique persistence-pairing of Edelsbrunner et al. is too restrictive to support a flexible hierarchy. Therefore, we use a more general pairing, which essentially pairs two nodes when they are connected by an arc. As the unique pairing is an integral part of the definition of persistence, we call \( p = |f(u) - f(v)| \) the variance of the pair \( u,v \).

2.3 Construction

In practice, we construct the MS complex by successively computing its arcs, starting from the saddles, see [3]. Starting from each saddle, we compute two lines of steepest ascent and two lines of steepest descent connecting the saddle to two maxima and two minima. We call these lines ascending and descending paths, directed always from the saddle towards an extremum. Two paths in the same direction (ascending or descending) can merge; two paths with different direction must remain separate. Once two paths have been merged they never split. Following these rules, we are guaranteed to produce a non-degenerate MS complex. A more detailed analysis can be found in [4]. Having computed all paths, we partition the surface into regions by growing each region from a triangle incident to a saddle without ever crossing a path.

2.4 Simplification

To simplify an MS complex locally we use a cancellation that eliminates two nodes. (The inverse operation to refine the complex is
called an anti-cancellation.) Only two nodes adjacent in an MS complex can be canceled. The possible configurations are a minimum and a saddle or a saddle and a maximum. Since the two cases are symmetric we limit our discussion to the second case, which is illustrated in Figure 3.

![Figure 3: Graph of a function before (a) and after (b) cancellation of pair \(u, v\).](image)

Let \(v\) be the saddle and \(u\) the maximum of the canceled pair, and let \(w\) be the other maximum connected to \(v\). We require that \(u \neq w\) and \(f(w) > f(v)\); otherwise, we prohibit the cancellation of \(u\) and \(v\). In particular, a cancellation or anti-cancellation must always maintain a valid MS complex. An MS complex is called valid, if all regions have four (not necessarily distinct) nodes and every path between a saddle and maximum/minimum is ascending/descending. Alternatively, an adaptively refined MS complex is valid if it can be created from the highest-resolution one using a sequence of cancellations.

3. CANCELLATION FOREST

In this section, we introduce the notion of a reduced MS complex as well as the cancellation forest. Furthermore, we show how a cancellation can be naturally separated into parts operating on the reduced complex and the cancellation forest. This separation results in a highly flexible encoding of cancellations.

3.1 Reduced MS Complex

The MS complex of a function, by construction, defines a quadrangulation of the underlying manifold. Furthermore, unlike typical surface meshes it also provides additional information about the topology of the function. Technically, this split into a surface mesh (describing the geometry of \(M\)) and functional information (describing the topology of \(f\)) is clear and well defined. Intuitively, however, a human usually splits the information differently as is apparent in many of the figures in this paper. Some figures like Figure 1 show the mesh aspect enhanced by node types. In such a drawing usually neither the exact position of the nodes nor their function values are considered important. Other figures, like Figure 17(a), illustrate the geometry of the surface as well as the function values but neither mesh nor topological structure. This separation into quadrangulation with node types and position and function values of nodes forms the core of our encoding. We call the quadrangulation with node types the reduced MS complex and the position and function values of all nodes the critical point configuration (CPC).

During cancellations and anticancellations the reduced complex changes more significantly, and incorrect encodings are easy to recognize. The top row of Figure 4 shows a reduced MS complex for three consecutive cancellations of minima \(C_1, C_2,\) and \(C_3\). To recover the correct reduced complex during an anticancellation one primarily needs connectivity information. For example, in Figure 4(d), \(m4\) must be created left of \(m3\) not right of \(m3\) a fact easily encoded by the neighborhood information, see Section 4. However, a reduced complex can be ambiguous. Reversing \(C1\) can result in the configuration shown in Figure 4(e) or (f), which are in fact identical reduced complexes drawn differently.

![Figure 4: MS complex (a) shown after three successive cancellations (b), (c), and (d). The configurations in (e) and (f) have the same connectivity but a different critical point configuration.](image)

3.2 Cancellation Forest

Separating mesh connectivity and vertex information is common, for example, in mesh compression. However, each coding/decoding step usually links both aspects directly. The easiest way to encode the CPC is to store its changes alongside the changes in the reduced complex: if a cancellation removed the critical point pair \(u, v\), then the corresponding anti-cancellation introduces \(u, v\). However, this imposes restrictions on the order of cancellations and anticancellations. Figure 5 shows the example of Figure 4 enhanced by labeling some nodes with function values. In this situation the configuration after reversing \(C1\) must be the one shown in Figure 5(c) and 4(f), respectively. The saddle \(s2\) cannot be connected to \(m0\) since the resulting path cannot be descending from saddle to minimum. However, \(C1\) removed \(s0, m1\), and linking the CPC directly to each cancellation would create an invalid MS complex. The algorithm proposed in [4] avoids these complications by imposing additional restrictions on the order of operations, see Section 4.

![Figure 5: MS complex of Figure 5 with function values. (a) Original complex. (b) Invalid critical point configuration path marked in red cannot be descending. (c) Valid critical point configuration requires anti-cancellation \(C1\) to create \(m2\) rather than \(m1\).](image)

We propose a different strategy that allows us to store the reduced complex and the CPC independently of each other using a simple data structure. The core idea is to view the cancellation shown in Figure 3 not as removing \(u\) and \(v\) but as merging the triple \(u, v,\) and \(w\) into \(w\). After a sequence of cancellations, we think of every extremum as the representative of itself plus all extrema (and saddles) merged with it. By definition, maxima only merge with maxima and minima only with minima. We keep track of these merge operations by creating a graph for every extremum: initially, each extremum is represented by itself as a graph with a single node. During each cancellation an arc is added between the corresponding graphs merging them into one. Since no extremum can
merge with itself, these graphs are in fact trees defining the cancellation forest. Figure 6 shows several cancellations and the resulting trees. Figure 13(a) shows part of the cancellation forest of a typical terrain data set.

![Cancellation forest example](image)

**Figure 6:** Example of a cancellation forest of maxima resulting from multiple cancellations. (Top) MS complex with some cancellations indicated in red. (Bottom) Corresponding trees of all maxima.

Even though the data structure used for the trees is simple, it is also very powerful due to two key properties. First, during a cancellation always the higher/lower maximum/minimum prevails in the MS complex. This fact implies that the representative of a tree of maxima/minima is always the highest/lowest node of the tree. In particular, the representative of each tree is uniquely defined. Second, arcs of a cancellation tree correspond to saddles that in turn correspond to cancellations. These two properties allow us to formulate a very simple bottom-up simplification procedure for MS complexes. Given a certain set of saddles that should be removed one can

1. build the corresponding cancellation forest by inserting an arc for every saddle;
2. determine the representative of each tree and remove all extrema that are not representatives; and
3. re-route every path to the representative of its original end point.

In step 1, saddles can be cancelled with either a minimum or a maximum unless they are part of a strangulation, see Figure 7, and connected to the same extremum twice. If there is a choice it depends solely on the simplification criterion used. For example, one could cancel the saddle with smallest variance, the one farthest away from the viewpoint, or use other criteria. The representative is the highest/lowest maximum/minimum of each tree and the re-routing is computed by concatenating original paths.

The procedure described above illustrates how useful a cancellation forest is during the simplification of an MS complex and defines a straightforward algorithm for off-line simplification. In practice, however, one is often interested in an interactively changing simplification, e.g., for a fly-over. Typically, a bottom-up strategy is too slow (especially for large data sets) and also provides no notion of gradual change necessary to avoid rendering artifacts. In applications that require even changing local adaptation, changes between successive simplifications are typically quite small and incrementally adapting one simplification to another is much faster than a bottom-up simplification.

4. HIERARCHY

To support incremental changes in a simplification we construct a hierarchy of (anti-)cancellations. Similarly to traditional hierarchies for polygonal meshes, operations are stored in a dependency graph representing a partial order among operations. We show how to encode a reduced MS complex in a hierarchical fashion and how the cancellation forest is used for incremental updates. First, we review the general data structure used to store an MS complex. Second, we compare the original hierarchy proposed in [4] to the new construction.

4.1 Hierarchy Construction

Following the approach discussed in [4], we split each region into two Morse triangles by introducing the diagonal connecting the minimum to the maximum into the complex. As a result, the neighborhood around a saddle consists of four triangles that form the diamond around the saddle, as indicated in grey in Figure 8(a).

![MS complex](image)

**Figure 8:** MS complex corresponding to Figure 3 (left) before and (right) after cancellation of pair \(u, v\). Diagonals indicating diamonds are shown as dotted lines.

Each cancellation removes one diamond from the (reduced) MS complex. We create a hierarchy in a bottom-up fashion by successively canceling critical points. Two cancellations are called independent if it is irrelevant in what order they are performed and dependent otherwise. The extended dependency graph contains a node for every cancellation and an arc between dependent cancellations. The dependency graph is derived from the extended one using path compression. The height of the dependency graph is defined as the maximal distance from a root to a leaf. In practice, one is interested in constructing a shallow graph with few arcs since this implies a large number of different configurations.

Clearly, the definition of dependencies between cancellations determines the shape of the dependency graph. In [4], the region of interference of the cancellation in Figure 8 is defined as all Morse cells incident to \(u, v,\) or \(w\). Two cancellations are defined as dependent if their regions of interference have a (true) intersection. This large region of interference is necessary to link the CPC information directly to each cancellation.

Given the large region of interference, storing the hierarchy is straightforward. Each cancellation replaces regions around three critical points by regions around the remaining one. The boundary of the region does not change, and the dependencies ensure that a (anti-)cancellation is only performed if the MS complex is locally identical to the one encountered during construction. This can be viewed as a special case of the concepts described for general multi-resolution structures described by de Floriani et al. [12]. Cancellations and the resulting dependency graphs using the old hierarchy are shown in Figure 9. Due to the large regions of interference the final dependency graph (lower-right corner) is a line allowing for no adaptations beyond the ones encountered during construction.
interference produce a smaller set of dependencies. In fact, one diamond plus its arc-connected neighbors. The smaller regions of neighborhood around a single diamond. Therefore, the region of interference of a cancellation is defined as the corresponding diamond plus its arc-connected neighbors again, but now with a significantly smaller region of interference. The smaller region of interference is shaded in red. The corresponding dependency graphs are shown next to the MS complexes. After four cancellations the dependency graph is a line. We propose a different approach by first concentrating solely on encoding the reduce complex. The reduced complex is stored by recording neighborhood relations among diamonds. Each cancellation removes one diamond replacing eight triangles around a vertex by four. An anti-cancellation re-introduces a diamond replacing four triangles by eight, introducing two vertices. Some possible configurations and their encodings are shown in Figure 10.

![Figure 9: Original hierarchy: Cancellations are indicated by arrows, the corresponding region of interference is shaded in grey, and regions of overlap with previous cancellations are shaded in red. The corresponding dependency graphs are shown next to the MS complexes. After four cancellations the dependency graph is a line.](image1)

Figure 10: Three examples for encoding a reduced MS complex. The complex (with diagonals) before (top) and after (bottom) a cancellation is shown. The middle row shows the neighborhood structure. A (anti-)cancellation is stored as a list of triangle pairs (-1 indicating a boundary arc).

To store cancellations hierarchically we use the concepts from [12] again, but now with a significantly smaller region of interference. In terms of the reduced complex, a cancellation changes only the neighborhood around a single diamond. Therefore, the region of interference of a cancellation is defined as the corresponding diamond plus its arc-connected neighbors. The smaller regions of interference produce a smaller set of dependencies. In fact, one sees that the number of ancestors and the number of children of each node in the resulting dependency graph is bounded (assuming path compression). One can imagine a cancellation not removing a diamond but rather collapsing it into a diagonal pair. The next cancellation involving either of these diagonals will become an ancestor, resulting in at most two ancestors. Since each diamond has four sides each cancellation has at most four children.

Given a corresponding cancellation forest a reduced MS complex is completed (to an MS complex) by assigning each generic extremum the representative of its tree as described at the end of Section 3. During each (anti-)cancellation the forest is iteratively updated. During each cancellation two cancellation trees are merged by adding the arc corresponding to the canceled saddle connecting them. The representative of the combined tree is the higher/lower maximum/minimum of the the two old representatives. Symmetrically, during an anticancellation an arc of the cancellation forest is deleted, and its corresponding saddle is added to the complex. The second node introduced by the anticancellation is given as the representative of one of the two cancellation trees just split. Overall, this strategy allows one to incrementally maintain the correct CPC without introducing additional dependencies to the hierarchy. Figure 11 shows the example of Figure 9 using the hierarchy of the reduced complex and cancellation trees.

![Figure 11: The top two rows show the example of Figure 9 using the cancellation forest to encode the hierarchy. The regions of interference are shaded in grey, and the corresponding extrema trees are drawn on the right side of each figure. Using the reduced MS complex all cancellations are independent. The bottom row shows the complex after the anti-cancellation of C1 (left) and C2 (right). Note that C1 cannot create M1 rather than M0, (M1 is higher than M0 as is evident by their cancellation order.](image2)

In terms of run time complexity, updating the reduced complex requires time linear in the valency of the nodes involved during cancellations as well as anticancellations. Updating the cancellation forest for a cancellation requires constant time (inserting an arc and comparing two nodes), and for an anticancellation requires time linear in the size of the corresponding cancellation tree (to search for one representative). Conceptually, cancellation trees bear a close resemblance to typical union-find structures, which would suggest a \( \log n \) search. However, cancellation trees can split at any arc at any time, which prohibits the use of a union-find structure. While more sophisticated structures are most likely possible our experiments suggest that the cancellation forest has an overall low branching factor, see 13(a). This diminishes any advantage of more complicated structures.

We create a hierarchy by removing diamonds from the highest-resolution MS complex in “batches” of independent cancellations. However, this strategy can result in cancellations of high persistence to be dependent on cancellations with much lower persistence, which is undesirable for most applications. Therefore, we limit the batches such that the largest persistence in a batch is not larger than twice the maximal persistence of the previous batch. Without this minor restriction, each batch contains at most four quarters of the remaining diamonds in the complex and therefore creates a hierarchy of logarithmic height.
To compare the new hierarchy with the one proposed in [4] we have applied both strategies to a 1201-by-1201 single-byte integer value terrain data set of the Grand Canyon. Figure 15 shows a rendering (a) and the initial MS complex (b) of the Grand Canyon data set with 11620 critical points. We assess quality via a fly-over, comparing the adaptivity of the original hierarchy with the one using the cancellation forest. A view-frustum is defined, where the topology is refined to the highest resolution. Outside the given view-frustum only dependent topology is used. Figure 16 shows two frames of the fly-over for two distinct stages of the fly-over path. An animation showing the complete fly-over can be downloaded at http://www.cipic.ucdavis.edu/~ptb/SMI05_FlyOver.mpg.

Figure 12: Number of critical points used during a fly-over (Grand Canyon data set).

The adaptive refinement and display of topology is useful in many areas. Figure 14 shows the oil pressure of an underground oil reservoir. (Oil is extracted by pressing water into the reservoir at some sites and pumping oil at others. As more water is forced into the reservoir it becomes increasingly saturated with water, and at some point oil production ceases to be effective.) The figure shows an isosurface of water saturation, pseudo-colored by oil pressure. The linear color map used in Figure 14(c) provides little structural information. However, the seven oil extraction sites are clearly visible as local minima in the simplified MS complex.

Figure 17(a) shows a rendering of the Yamka data set using 1201 × 1201 single-byte integer values, (b), and (c) show the corresponding MS complex with 17691 critical points and the same complex refined to preserve only features below a function value of 0.14 (with function values scaled to [0, 1]) using 8063 critical points. The density of the MS complex shows how the region around the canyons remains highly refined.

Figure 18 shows the Mixing Fluid data set. The surface is an isosurface representing the boundary between two mixing fluids extracted from one time-step of a turbulent mixing simulation. The data has been generated by the Miranda code a higher order hydrodynamics code for computing fluid instabilities and turbulent mixing at the Lawrence Livermore National Laboratory [7]. In particular, scientists are interested in "bubbles" formed during the mixing process and their automatic segmentation. Using the z-coordinate as Morse function on the surface bubbles are described by the descending manifolds of maxima as shown in Figure 18. Nevertheless, the segmentation of Figure 18(a) is not optimal as some bubbles have multiple maxima and there exist many superfluous maxima cause by noise in the data set. Using a uniform simplification of the MS complex one can remove most of these artifacts and create a much cleaner segmentation, as shown in Figure 18(b). Figure 18(c) and (d) show a non-uniform refinement of the same data which concentrates only on data within the grey focus sphere. As in the Grand Canyon fly-over, the hierarchy using the cancellation forest proves to be far more adaptive than the original one of [4].


Figure 13: Typical extrema trees of a terrain data set. Maxima are shown in red, minima in blue, and arcs in green. Note the overall low branching factor.

Figure 14: Pseudo-colored rendering and simplified MS complex of oil-pressure data set.
Figure 15: (a) Rendering of Grand Canyon data set; (b) original MS complex of (a) using 11620 critical points (minima shown in blue, maxima in red, and saddles in green.)

Figure 16: Two frames of the fly-over of the Grand Canyon data set. (Left) Using the original hierarchy; (right) using the cancellation forest.
Figure 17: (a) Rendering of the Yakima data set; (b) original MS complex of (a) (17691 critical points); (c) adaptively refined MS complex, where only features below function value of 0.14 are preserved (8063 critical points).

Figure 18: Color-mapped rendering of the Mixing Fluids data set including critical points and descending arcs. (a) Highest resolution with 252 maxima; (b) simplified data with 140 maxima showing one maximum for each "bubble." The bottom row shows the same data set locally refined to only preserve critical points within the focus sphere shown in grey. (c) Using the hierarchy of [4] with 155 maxima; (b) using the cancellation forest with 80 maxima.