# Maximizing Adaptivity in Hierarchical Topological Models Using Extrema Trees

Peer-Timo Bremer[1], Valerio Pascucci[2], and Bernd Hamann[3]

[1] Department of Computer Science, University of Illinois, Urbana-Champaign
ptbremer@acm.org
[2] Center for Applied Scientific Computing, Lawrence Livermore National Laboratory
pascucci@llnl.gov
[3] Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis hamann@cs.ucdavis.edu

**Summary.** We present a highly adaptive hierarchical representation of the topology of functions defined over two-manifold domains. Guided by the theory of Morse-Smale complexes, we encode dependencies between cancellations of critical points using two independent structures: a traditional mesh hierarchy to store connectivity information and a new structure called extrema tree to encode the configuration of critical points. Extrema trees provide a powerful method to increase adaptivity while using a simple, easy-to-implement data structure. The resulting hierarchy is 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 iso-contouring, unless "correct" transfer functions and isovalues 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 in 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 preform incremental changes for varying queries.

We address the need for adaptive topology-based data exploration by improving significantly the topological hierarchy proposed in [4]. Creating two largely independent hierarchies, 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 [17, 5, 18], 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 [24]. However, the contour tree provides little information about the embedding of the level sets and therefore remains somewhat abstract. Morse theory [16, 15], 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 bivariate case are provided in [6, 14, 19]. More recently, the Morse-Smale complex was introduced by Edelsbrunner et al. [9, 8] 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 [21] to shape description [13]. Related concepts are also used in flow visualization. Helman and Hesselink [12] showed how to find and classify critical points in flow fields and propose 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 [7] and Tricoche et al. [22, 23].

The first multi-resolution encoding of a Morse-Smale complex we are aware of was proposed by Pfaltz [20], which has been improved and extended by Edelsbrunner et al. [9] and Bremer et al. [3, 4]. More recent hierarchical structures are based on the concept of *persistence* [10], which relates the difference in function value of critical point pairs to the importance of a topological feature. Given a Morse-Smale complex, we

1. provide an improved hierarchical encoding of the Morse-Smale complex;
2. prove that the resulting hierarchy is of logarithmic height; and
3. demonstrate our methods for various 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 extrema trees and 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 [16, 15] for smooth functions and discuss how these can be used in the piecewise linear case.

### 2.1 Morse Theory

Given a smooth function $f : \mathbb{M} \to \mathbb{R}$, a point $a \in \mathbb{M}$ is called *critical* when its *gradient* $\bigtriangledown f(a) = (\delta f/\delta x, \delta f/\delta y)$ 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 $a$ and 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 [9], we can overlay these two complexes and obtain what we call the *Morse-Smale complex*, or MS complex, of $f$. Its vertices are the vertices of the two overlayed complexes, which are the minima, maxima, and saddles of $f$. Its cells are four-sided regions bounded by parts of integral lines between saddles and extrema. An example is shown in Figure 1.

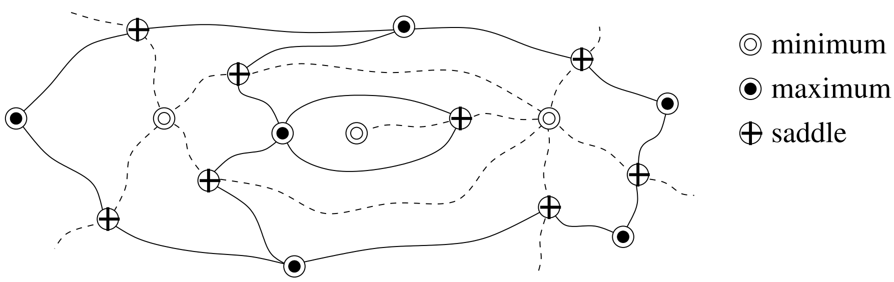



**Fig. 1.** Morse-Smale complex.

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, 9]. If all vertices that are edge-connected to a point $u$ have function values below that of $u$, we call it a maximum; if all are above $u$, then we call it a minimum etc., see Figure 2. In general, there can exist saddles with high multiplicity that we split into simple ones, as shown on the far right in Figure 2.

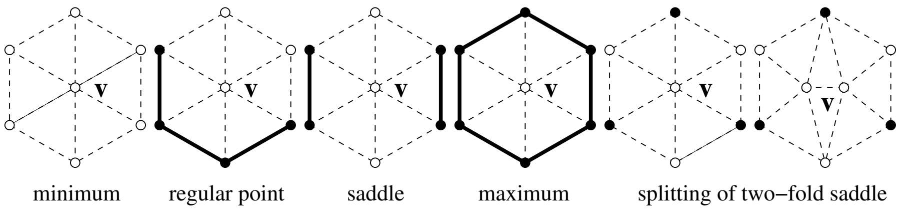



**Fig. 2.** Classification of a vertex v based on relative height of its edge-connected neighbors, where light vertices/edges mark higher neighbors and solid vertices/edges lower neighbors.

### 2.2 Persistence

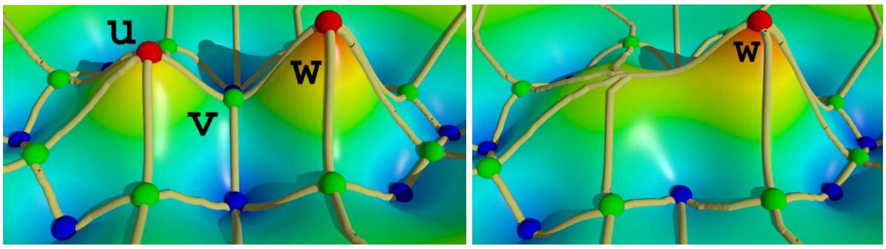
As a numerical measure of the importance of critical points we define pairs of critical points and use the absolute difference between their height/function values. The underlying intuition is the following: We imagine sweeping the two-manifold $\mathbb{M}$ in the direction of increasing height (w.r.t. the scalar field value.) The topology of the part of $\mathbb{M}$ below the sweep line 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. We pair a vertex $v$ that creates a component with the vertex $u$ that destroys the component. The *persistence* of $u$ and of $v$ is the "delay" between the two events: $p = f(v) - f(u)$, see [10].

### 2.3 Construction

In practice, we construct the MS complex by successively computing its edges, 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* or *descending paths*. 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 [3]. Having computed all paths, we partition the surface into four-sided 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.

### 2.4 Simplification

To simplify an MS complex locally we use a *cancellation* that eliminates two critical points. The inverse operation to refine the complex is called an *anti-cancellation*. Only two critical points 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.




(a) (b)

**Fig. 3.** Graph of a function before (a) and after (b) cancellation of pair $u, v$.

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 cells have four (not necessarily distinct) corners 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 Extrema Trees

The information an MS complex provides can be separated into the critical points and their connectivity. The critical points information includes position, type, and function value and we refer to this as *critical point configuration (CPC)*. The connectivity encodes which paths (edges) define a Morse cell and the neighboring information between cells. As with most mesh encoding schemes the CPC provides most (but not all) information about the MS complex. Especially during simplification, the connectivity of the MS complex can often be infered from the CPC. For example, in Figure 3 after $u$ and $v$ have been removed all saddles that were connected to $u$ are now connected to $w$.

When encoding a cancellation the separation between CPC and connectivity is very intuitive. The top row of Figure 4 shows three consecutive cancellations $C1$, $C2$, and $C3$ of minima. To reverse any of these cancellations one first needs to know how the connectivity of the MS complex changes. For example, in Figure 4(d) $m4$ must be created on the left of $m3$ (not on its right.) This information is provided by the neighborhood relations between Morse cells, see Section 4. However, knowing the appropriate connectivity is still leaving ambiguity. Reversing $C1$ seems to result in the structure of Figure 4(e). Nevertheless, the MS complex drawn in (f) has the same connectivity but a different CPC.

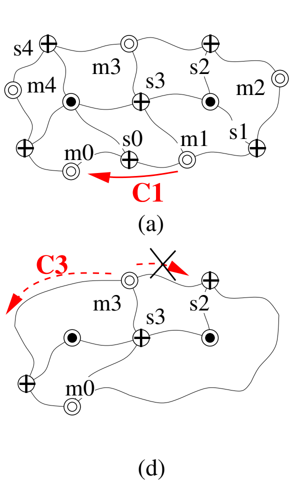
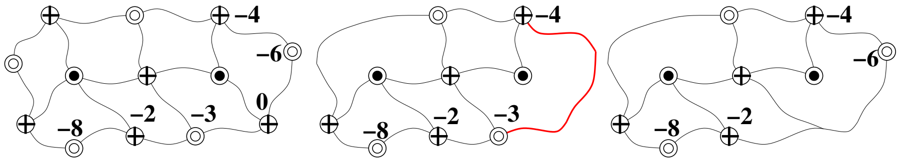



**Fig. 4.** MS complex (a) shown after three successive cancellations (b), (c), and (d). The configurations in (e) and (f) have the same connectivity but a different critical point configuration.

The straightforward solution to encoding the CPC is to link it directly to each cancellation. If a cancellation removed the critical point pair $u, v$ then the corresponding anti-cancellation would introduce $u, v$. However, this imposes restrictions on the order of cancellations and anti-cancellations. Figure 5 shows the example of Figure 4 enhanced by labeling some critical points 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 could not 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.

We propose a different strategy that allows us to store connectivity and 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 merged with it. Maxima only merge with maxima and minima only with minima. We keep track of these merges by creating a graph for every extremum. Initially, each extremum is represented by itself as a




(a) (b) (c)

**Fig. 5.** MS complex of Figure 5 with function values. (a) Original complex. (b) Invalid critical point configuration (the path marked in red cannot be descending.) (c) Valid critical point configuration requires anti-cancellation $C1^{-1}$ to create $m2$ rather than $m1$.

graph with a single node. During each cancellation an edge is added between the corresponding graphs merging them into one. Since no extremum can merge with itself these graphs are trees, called *extrema trees*. Figure 6 shows several cancellations and the resulting extrema trees. Figure 13(a) shows the extrema trees of a typical terrain data set.

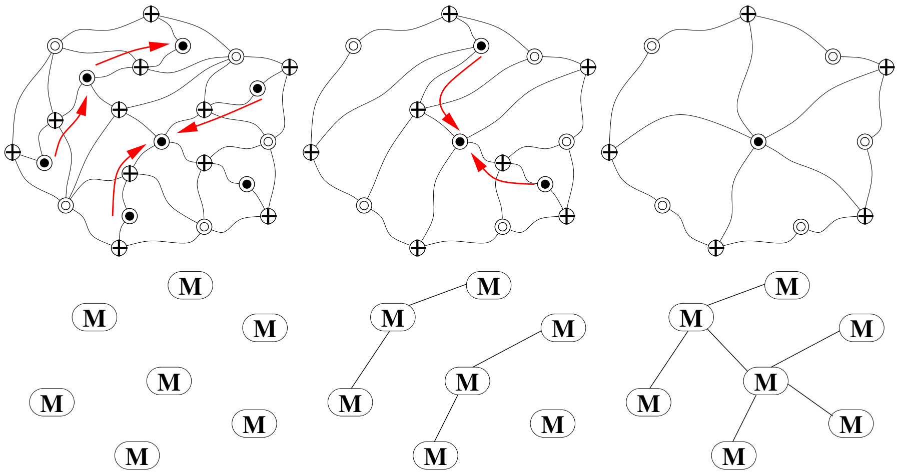


**Fig. 6.** Example of extrema trees of maxima resulting from multiple cancellations. (Top) MS complex with some cancellations indicated in red. (Bottom) Corresponding extrema trees of all maxima.

Even though the data structure used for extrema trees is simple, it is also very powerful due to two key properties. First, recall that 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. Second, arcs of an extrema tree correspond to saddles and/or cancellations. In fact, given some extrema trees, it is possible to derive a (nearly) complete MS complex based only on a set of saddles. Assume one is given a highly simplified MS complex and a set of extrema trees; furthermore, assume a (local) refinement of the MS complex is given by a set of saddles $S = \{s_0, \ldots, s_n\}$ that must appear in the refined complex. First, one removes all arcs corresponding to a saddle in $S$ from the extrema trees. Subsequently, one can reconstruct the MS complex in the following manner: Each saddle $s_i$ was initially connected to two maxima $M_0, M_1$ and two minima $m_0, m_1$. All of these extrema are part of a tree, and the saddle is connected to the four representatives of these trees. This defines the adaptive MS complex to the level of the embedding of the paths. The saddles are given, the remaining critical points are the representatives of the extrema trees, and the paths embedding can be derived from concatenating original paths.

Nevertheless, the connectivity between Morse cells is not uniquely defined by the construction described above. This is due to the fact that in an MS complex paths are not uniquely defined by their end points, see Figure 7. As a result, Morse cells are not identified by their corners and the connectivity must still be stored explicitly. Section 4 describes how the connectivity as well as the configuration of saddles can be stored hierarchically. The extrema trees are used to complete the CPC.

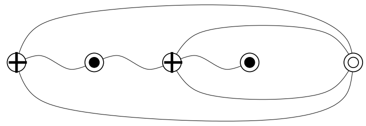


**Fig. 7.** Strangulation where two Morse cells have the same corners.

Maintaining extrema trees is a constant-time operation during a cancellation and involves a linear search during an anti-cancellation. In general, an extrema tree can be split anywhere at any time. This prohibits the use of standard acceleration structures such as a union-find approach. While more sophisticated structures are possible our experiments suggest that extrema trees have an overall low branching factor. This diminishes any advantage of more complicated structures and would make the implementation more difficult.

## 4 Hierarchy

Using extrema trees to maintain the CPC allows us to create a mesh hierarchy geared completely towards connectivity. The main objective is to construct a hierarchy that supports as many different configurations as possible. Similarly to traditional hierarchies for polygonal meshes, (anti-)cancellations are stored in a dependency graph representing a partial order among operations. All configurations that can be created by observing the partial order should result in a valid MS complex.

### 4.1 Hierarchy Construction

Following the approach discussed in [4], we split each Morse cell 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 then consists of four triangles that form the *diamond* around the saddle, as indicated in grey in Figure 8(a). Each cancellation removes one diamond from the MS complex. We create a hierarchy in a bottom-up fashion by successively canceling critical points. Two cancellations are

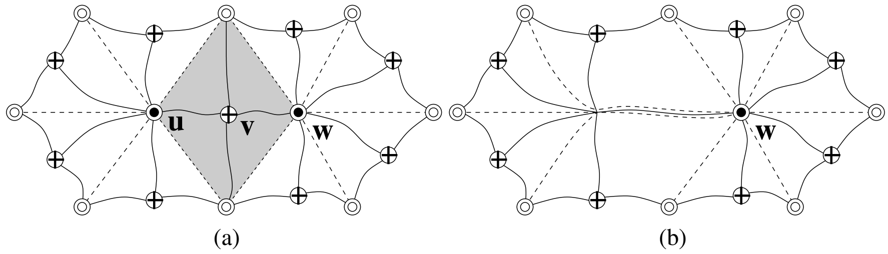



**Fig. 8.** MS complex corresponding to Figure 3 (a) before and (b) after cancellation of pair $u,v$. Diagonals indicating diamonds are shown as dotted lines.

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 edges since this implies the possibility of 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 either $u$, $v$, 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 avoid the problems discussed in Section 3. Given the large region of interference, storing the hierarchy is straightforward. Each cancellation replaces Morse cells around three critical points by Morse cells 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, for example, by de Floriani et al. [11]. An example of several cancellations and the resulting dependency graphs using the old hierarchy is shown in Figure 9. Due to the large regions of interference the final dependency graph (lower right corner) is a line allowing no adaptations beyond the ones encountered during construction.

Using extrema trees one can ignore the configuration of minima and maxima, requiring us to encode only the connectivity and saddle configuration. Since each cancellation removes the diamond around a saddle it is natural to link the saddle information directly to a diamond. Therefore, if we can store the diamond information (the connectivity) hierarchically, extrema trees provide the remaining information.

To store the connectivity information we use the concepts from [11] but now with a significantly smaller region of interference. Each cancellation removes one diamond replacing eight triangles around a vertex by four. An anti-cancellation reintroduces a diamond replacing four triangles by eight, introducing two vertices. Some possible configurations are shown in Figure 10. The cancellation of a dia-

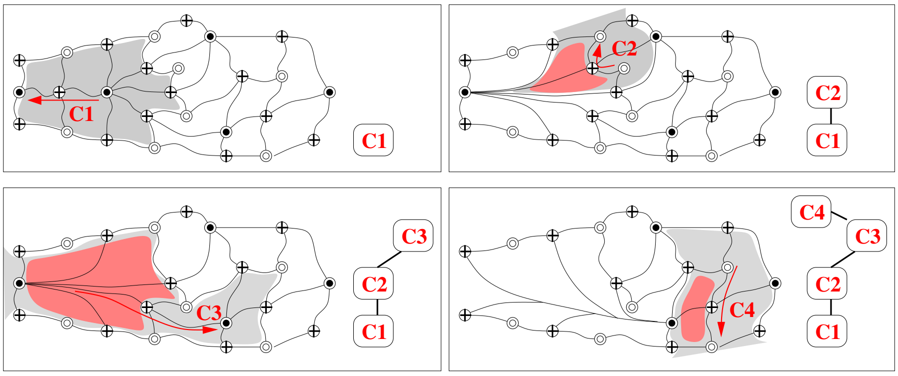



**Fig. 9.** Hierarchy construction as described in [4]. Cancellations are indicated by arrows, the corresponding region of interference is shaded in grey, and regions of overlap with previous cancellations are shaded in red. The corresponding dependency graphs are shown next to the MS complexes. After four cancellations the dependency graph is a line.

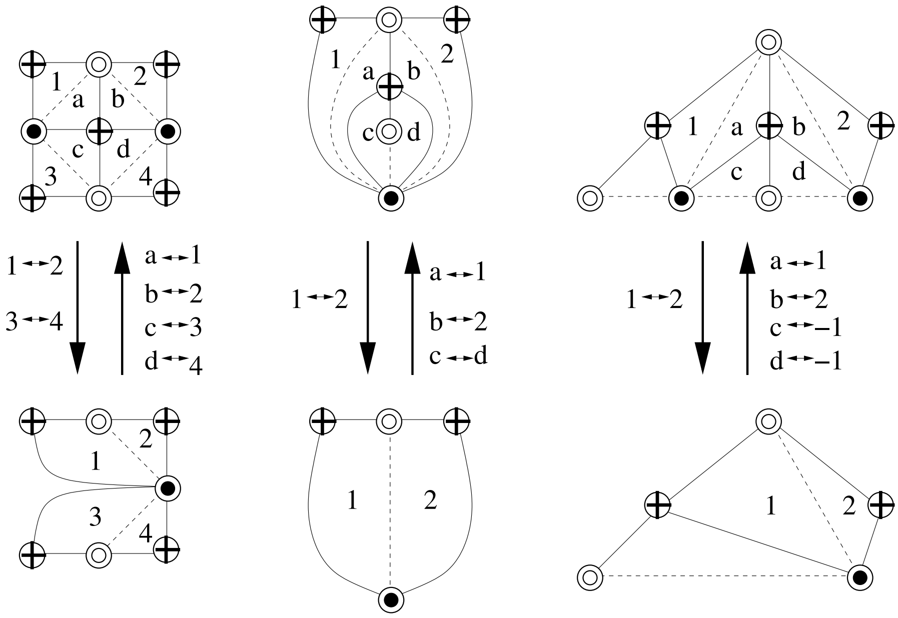



**Fig. 10.** Three examples for encoding the connectivity during cancellations. The triangulation before (top) and after (bottom) a cancellation is shown. The middle row shows how the neighborhood structure for a cancellation. An anti-cancellation is stored as a list of triangle pairs (-1 indicating a boundary edge).

mond changes a reduced MS complex only for the neighboring (edge-connected) diamonds. Therefore, the region of interference of a cancellation is defined as the corresponding diamond plus its edge-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 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. Each cancellation has at most four children. Figure 11 shows the example of Figure 9 using extrema trees.

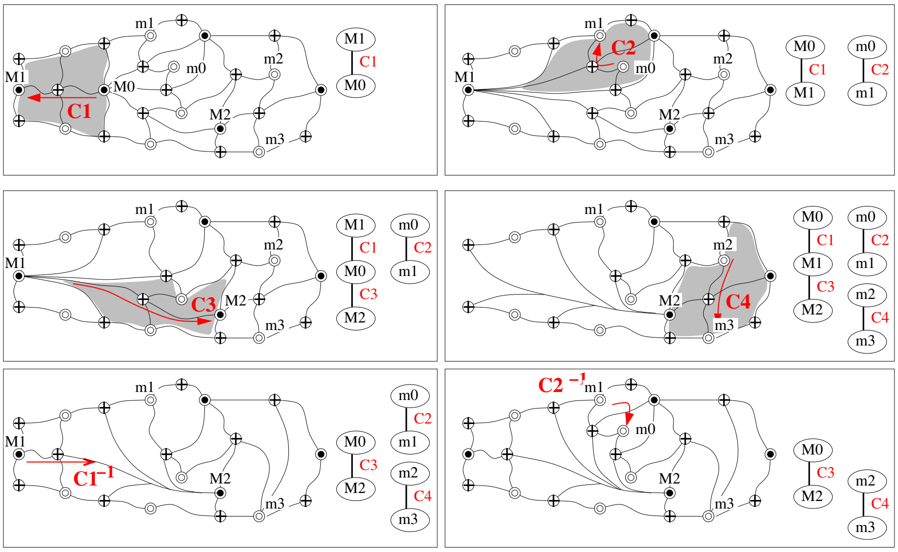


**Fig. 11.** The top two rows show the example of Figure 9 using extrema trees to encode the hierarchy. The regions of interference are shaded in grey, and the corresponding extrema trees are drawn on the right side of each figure. Using the reduced MS complex all cancellations are independent. The bottom row shows the complex after the anti-cancellation of $C1$ (left) and $C2$ (right). Note that $C1^{-1}$ correctly creates $M1$ rather than $M0$ ($M1$ is higher than $M0$).

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 about one quarter of the remaining diamonds in the complex and therefore creates a hierarchy of logarithmic height.

## 5 Results

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 16 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 cell-based hierarchy with the one using extrema trees. A narrow view-frustum is defined where the topology is refined to the highest resolution. Outside the given view-frustum only dependent topology is used. Figures 17 and 18 show two frames of the fly-over for two distinct stages of the fly-over path.

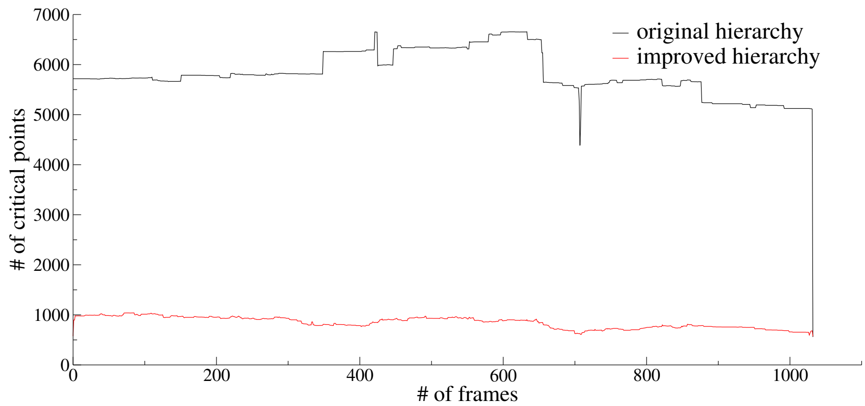



**Fig. 12.** Number of critical points used during a fly-over (Grand Canyon data set.)

Figure 12 shows the number of critical points in the adaptive MS complex during the fly-over for both methods used for hierarchy construction. The hierarchy using extrema trees is clearly superior to the original encoding. One explanation for the large differences in quality is the presence of high-valency extrema in the MS complex. Often, data sets (especially terrains) are biased to contain significantly more maxima than minima (or the reverse), which consequently results in some extrema of the MS complex with high valency values. Using the original large region of interference, the hierarchy around a high-valency extremum degenerates into a linear sequence.

The adaptive refinement and display of topology is useful for many areas. Figure 15 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 the reservoir 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 15(c) provides little structural information. However, the seven oil extraction sites are visible as local minima in the simplified MS complex.

Figure 13(b) shows a rendering of the Yakima data set using $1201 \times 1201$ single-byte integer values, and Figure 14 shows 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.

## 6 Conclusions and Future Research

We have improved our original results discussed in [3] significantly in several different ways, moving towards the practical application of topology for data visualization and analysis. Using extrema trees, the hierarchy is smaller, more adaptable, and supports the use of larger, more complicated MS complexes. Further, extrema trees are easy to implement and maintain during refinement. One disadvantage of the new technique is that the hierarchy is so flexible that it becomes impossible to precompute function values corresponding to all possible topological refinements. Therefore, only the adapted topology, not the corresponding adapted function, can be displayed interactively. We plan to develop new techniques computing high-quality topological approximation on-the-fly.

## 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. B. Hamann is supported by National Science Foundation under contract ACI 9624034 (CAREER Award), through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251, through the National Partnership for Advanced Computational Infrastructure (NPACI), and through a large Information Technology Research (ITR) grant. We thank the members of Data Science thrust from the Center for Applied Scientific Computing (CASC) at Lawrence Livermore National Laboratory, and the Visualization and Computer Graphics Research Group at the University of California, Davis.

## References

1. C. L. Bajaj, V. Pascucci, and D. R. Schikore. Visualization of scalar topology for structural enhancement. In D. Ebert, H. Hagen, and H. Rushmeier, editors, *Proc. IEEE Visualization '98*, pages 51–58, Los Alamitos California, 1998. IEEE, IEEE Computer Society Press.
2. T. F. Banchoff. Critical points and curvature for embedded polyhedral surfaces. *American Mathematical Monthly*, 77(5):457–485, May 1970.
3. P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A multi-resolution data structure for two-dimensional Morse-Smale functions. In G. Turk, J. J. van Wijk, and R. Moorhead, editors, *Proc. IEEE Visualization '03*, pages 139–146, Los Alamitos California, 2003. IEEE, IEEE Computer Society Press.
4. P.-T. Bremer, H. Edelsbrunner, B. Hamann, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 10(4):385–396, 2004.

5. H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Comput. Geom. Theory Appl.*, 24(3):75–94, 2003.
6. A. Cayley. On contour and slope lines. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XVIII:264–268, 1859.
7. W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *Proc. IEEE Visualization '99*, pages 349–354. IEEE Computer Society Press, 1999.
8. H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Morse-Smale complexes for piecewise linear 3-manifolds. In *Proc. 19th Sympos. Comput. Geom.*, pages 361–370, 2003.
9. H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Comput. Geom.*, 30:87–107, 2003.
10. H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.
11. L. De Floriani, E. Puppo, and P. Magillo. A formal approach to multiresolution modeling. In W. Straßer, R. Klein, and R. Rau, editors, *Theory and Practice of Geometric Modeling*. Springer Verlag, 1996.
12. J. L. Helman and L. Hesselink. Visualizing vector field topology in fluid flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May/Jun. 1991.
13. M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In E. Fiume, editor, *Proceedings of ACM SIGGRPAH 2001*, pages 203–212, New York, NY, USA, 2001. ACM.
14. J. C. Maxwell. On hills and dales. *The London, Edinburgh and Dublin Philosophical Magazine and Journal of Science*, XL:421–427, 1870.
15. J. Milnor. *Morse Theory*. Princeton University Press, New Jersey, 1963.
16. M. Morse. Relations between the critical points of a real functions of n independent variables. *Transactions of the American Mathematical Society*, 27:345–396, July 1925.
17. S. P. Morse. A mathematical model of the analysis on contour-line data. *Journal of the Association for Computing Machinery*, 15(2):205–220, Apr. 1968.
18. V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In M. Gross, K. I. Joy, and R. J. Moorhead, editors, *Proc. IEEE Visualization '02*, pages 187–194, Los Alamitos California, 2002. IEEE, IEEE Computer Society Press.
19. J. Pfaltz. Surface networks. *Geographical Analysis*, 8:77–93, 1976.
20. J. Pfaltz. 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*, no. 73, 1979.
21. B. T. Stander and J. C. Hart. Guaranteeing the topology of implicit surface polygonization for interactive modeling. In *Proc. of ACM SIGGRPAH 1997*, volume 31, pages 279–286, New York, NY, USA, Aug. 1997. ACM, ACM Press / ACM SIGGRAPH.
22. X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2d vector fields. In *Proc. IEEE Visualization '00*, pages 359–366, Los Alamitos California, 2000. IEEE, IEEE Computer Society Press.
23. X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proc. IEEE Visualization '01*, pages 159–166, Piscataway, NJ, Oct. 2001. IEEE, IEEE Computer Society Press.
24. M. J. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Symposium on Computational Geometry*, pages 212–220, 1997.

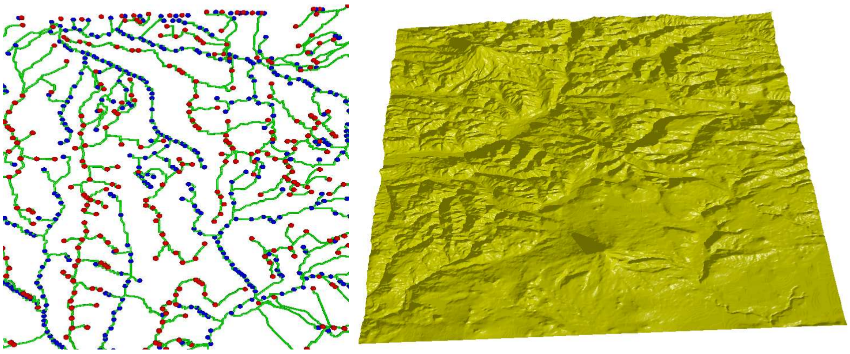


**Fig. 13.** (Left) Typical extrema trees of a terrain. Maxima are shown in red, minima in blue, and arcs in green. Note the overall low branching factor. (Right) Rendering of original Yakima data set.

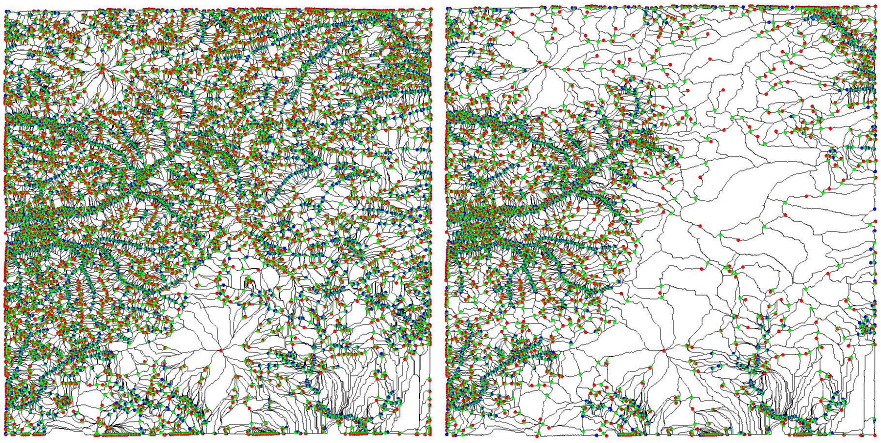


**Fig. 14.** (Left) Original MS complex of the Yakima data set (17691 critical points); (right) adaptively refined MS complex, where only features below function value of 0.14 are preserved (8063 critical points).

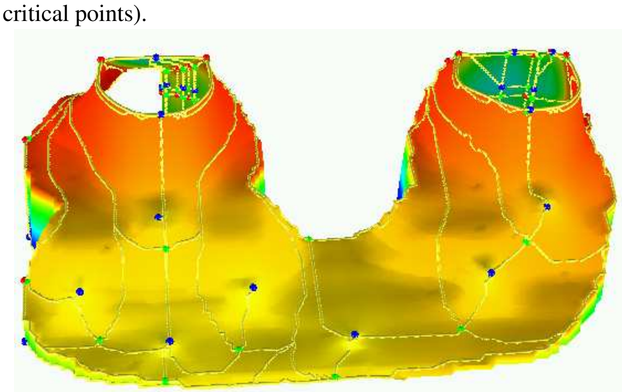
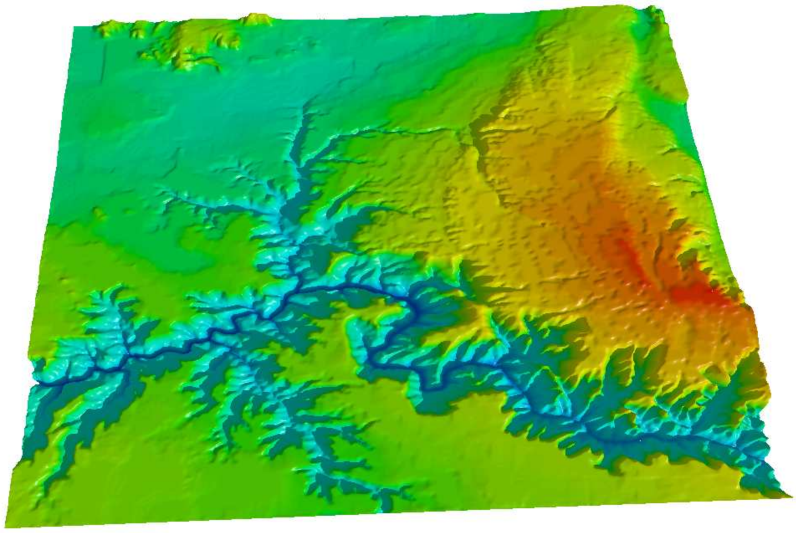


**Fig. 15.** Pseudo-colored rendering and simplified MS complex of oil-pressure data set.



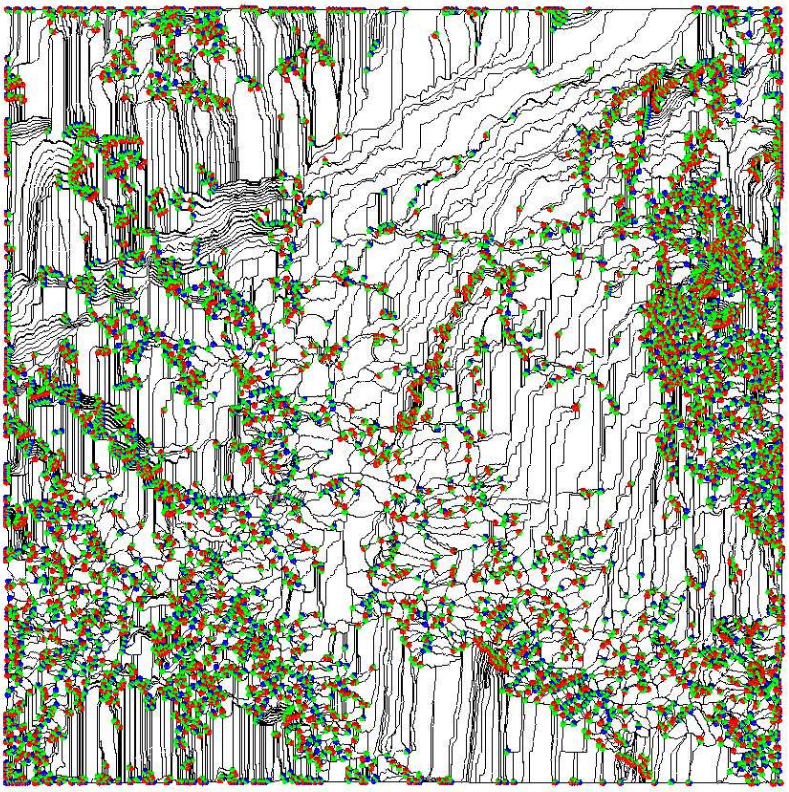


**Fig. 16.** Rendering of Grand Canyon data set; (b) original MS complex of (a) using 11620 critical points (minima shown in blue, maxima in red, and saddles in green.)

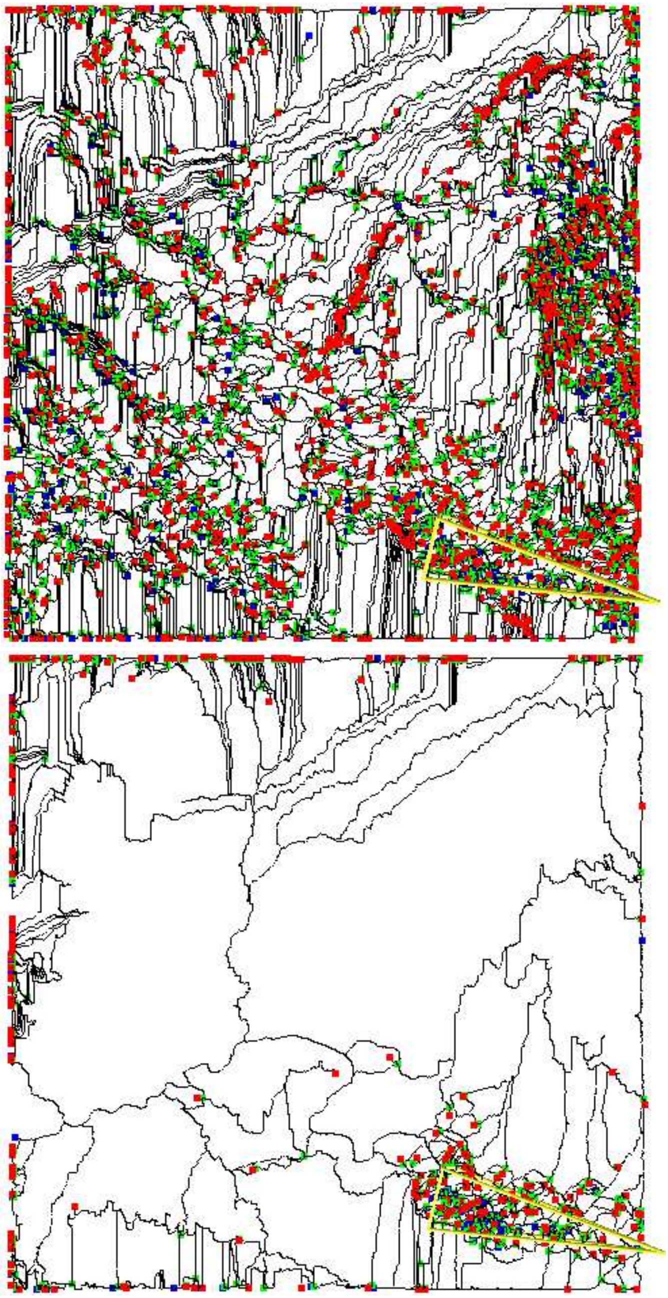


**Fig. 17.** Global view of a fly-over of Grand Canyon data set. Inside the local view frustum (yellow) the finest resolution topology is shown on the outside only dependent topology is used. (Top) The results of the original hierarchy; (bottom) refinement using the improved hierarchy introduced in this paper.

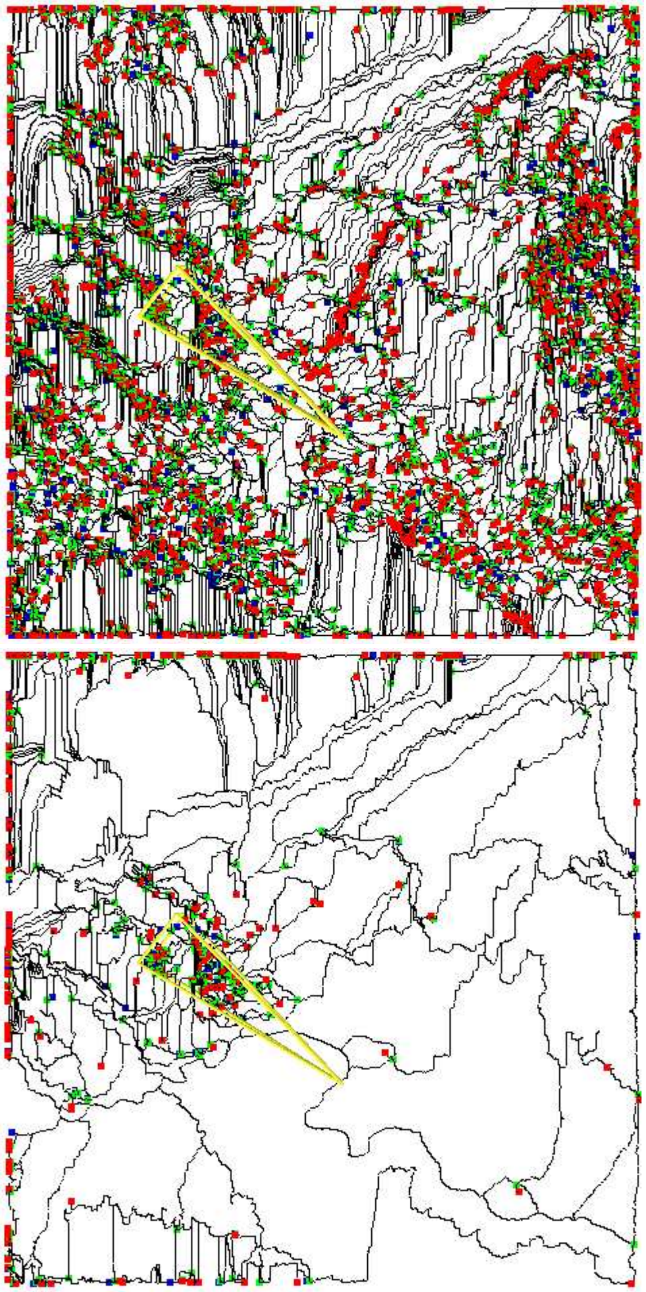


**Fig. 18.** Another frame of the fly-over of the Grand Canyon data set. (Top) Using the original hierarchy; (bottom) using extrema trees.