Contour Tree Simplification With Local Geometric Measures

Hamish Carr
Dept. of Computer Science
Univ. College Dublin

Jack Snoeyink
Dept. of Computer Science
UNC Chapel Hill

Michiel van de Panne
Dept. of Computer Science
Univ. of British Columbia

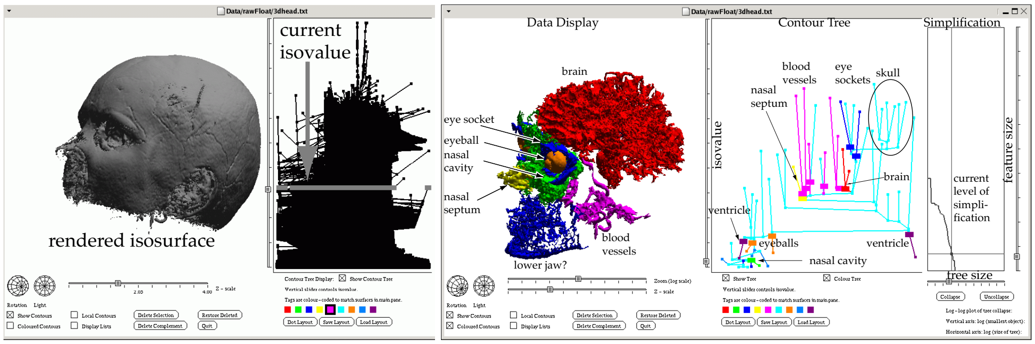

Figure 1: Left, an isosurface of the UNC Head ($109 \times 256 \times 256$ MRI) shows mostly the skull: the contour tree is unmanageable (1,573,373 edges). Right, contour surfaces chosen using a simplified contour tree. (Annotation and colour chosen to emphasize the structure of the data.)

Abstract

The contour tree, an abstraction of a scalar field that encodes the nesting relationships of isosurfaces, has several potential applications in scientific and medical visualization, but noise in experimentally-acquired data results in unmanageably large trees. We attach geometric properties of the contours to the branches of the tree and apply simplification by persistence to reduce the size of contour trees while preserving important features of the scalar field.

Keywords: Isosurfaces, contour trees, topological simplification

1 Introduction

The *contour tree* is a topological abstraction of a scalar field used in scientific and medical visualization [BPS97; vKvOB+97; PCM02; CSvdP04]. It represents changes in isosurface connectivity. In this paper, we simplify the contour tree using geometric properties of contours, permitting online simplification of the contour tree.

Figure 1 shows a conventional isosurface and a flexible isosurface [CS03] extracted from the same data set after contour tree simplification. On the left, we see that the outermost surface (the skull) occludes other surfaces, making it difficult to study structures inside the head, and the contour tree has too many edges to be useful. On the right, we see the result of using the simplified contour tree as an interface tool to enable a user to explore, color, and annotate the contours – the structures inside the head can be seen in relation to each other.

2 Related Work

The contour tree, a special case the Reeb graph [Ree46], is the result of contracting each contour in a scalar field to a single point; it tracks how *contours*, connected components of isosurfaces of a data set, appear, merge, split, and vanish as we vary the chosen isovalue. Efficient algorithms for constructing the contour tree have been reported for various meshes and interpolants [vKvOB+97; CSA03; PCM02; CLLR02; TNTF04]; the contour tree has applications ranging from fast isosurface extraction [vKvOB+97; CS03] and volume rendering [TNTF04] to mesh simplification, abstract representation of scalar fields [BR63; BPS97] and contour manipulation [CS03]. Unfortunately, noise in the input data can create many new contours by creating local minima and maxima. For noisy experimentally-acquired data such as the UNC head data set shown in Figure 1, contour trees commonly have millions of edges – too many to serve as a visual representation for the input data.

To simplify the contour tree, we would like to assign an importance to each edge and collapse edges of lower importance. This is a simple case of the ideas of *topological persistence* [EHZ03; ELZ02; BEHP03] applied to trees. Two works have applied persistence to the isovalues: [HSKK01] simplify the Reeb graph using hierarchical quantization of the data values, which can introduce errors at edges that span the quantization boundaries, and [TNTF04] simplify the contour tree using data values. We allow any geometric property to guide simplification (and are most efficient with decomposable properties, such as volume and surface area.)

3 Contour Tree Simplification

Given a contour tree and a scalar field, we simplify the contour tree with graph operators, then reflect the simplication back to the input data or use the simplified contour tree directly to extract individual contours from the simplified data set.

To compute geometric measures for individual contours, we replace the single isovalued sweep in [BPS97] with multiple separate sweeps of individual contours corresponding to sweeping individual points through the tree. Doing this requires combining partial sweep results whenever a saddle in the tree is swept past. In three dimensions, we can compute surface area, volume or *hypervolume*: isovalue integrated inside the contour.

To simplify the contour tree, we then choose a leaf edge that corresponds to contours for which the chosen geometric property is small and *prune* the leaf from the tree. By removing only leaves, we guarantee that the structure remains a tree and corresponds to a subregion of the scalar field in which isovalued contour sweeps can still be performed without discontinuous jumps.

Leaf pruning can result in a redundant vertex in the tree, as in Figure 2. We remove such vertices with no loss of topological information in the tree. Since there is no geometric cost to doing so, we prefer these vertex simplications where available and also prefer leaf prunes that maximize the number of future vertex simplifications.

Removing a leaf of the tree corresponds to flattening a local extremum of the data set as shown in Figure 2. By minimizing the geometric cost of our simplification, we are able to achieve simplification of the tree by 4 orders of magnitude without causing significant errors in the underlying field being represented, and preserving details of the contours that remain.

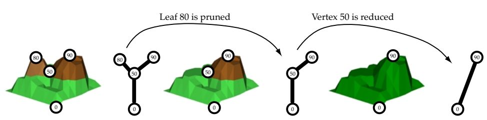

Figure 2: Leaf Pruning Levels Extrema; Vertex Reduction Leaves Scalar Field Unchanged

4 Results and Discussion

We have tested this form of simplification on a variety of data sets using the flexible isosurface interface [CS03]. In Figure 1, we show a typical result using hypervolume as the importance measure. Contours for the skull were not selected because they occlude the internal organs.

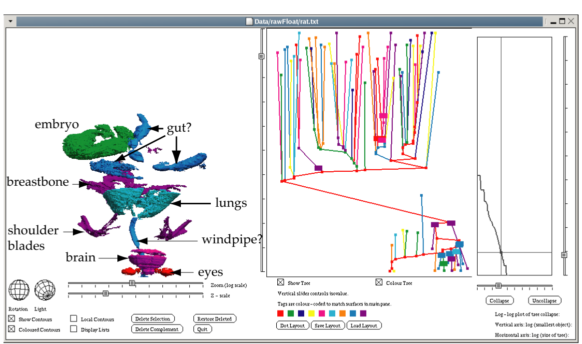

Figure 3: A Pregnant Rat MRI ($240 \times 256 \times 256$). Despite low quality data, simplifying the contour tree from 2,943,748 to 125 edges allows identification of several anatomical features.

Similarly, Figure 3 shows the result of a similar exploration of a $240 \times 256 \times 256$, low-quality MRI scan of a rat from the *Whole Frog Project* at `http://www-itg.lbl.gov/ITG.hm.pg.docs/Whole.Frog/Whole.Frog.html`. Again, simplification reduces the contour tree to a useful size. After using the *dot* tool from the graphviz package (`http://www.research.att.com/sw/tools/graphviz/`) to lay out the contour tree, these images took less than 10 minutes to explore and annotate. The result is purely a function of the topology of the isosurfaces of the input data, and uses no special constants.

5 Conclusions and Future Work

We have presented a novel algorithm for the simplification of contour trees based on local geometric measures. The algorithm is *online*, meaning that simplifications can be done and undone at any time. This addresses the scalability problems of the contour tree in exploratory visualization of 3D scalar fields. The simplification can also be reflected back onto the input data to produce an on-line simplified scalar field. The algorithm is driven by local geometric measures such as area and volume, which make the simplifications meaningful. Moreover, the simplifications can be tailored to a particular application or data set.

Future directions of research include extension to vectors of geometric measures, user-directed local simplification of the contour tree, utilization of the contour tree as a query structure for geometric properties, application of similar methods to volume rendering and to non-isovalue segmentation, extension to time-varying data sets, parallelization and improvements to contour tree layout algorithms.

6 Acknowledgements

Acknowledgements are due to NSERC, NSF and IRIS for funding, and to the sources of volumetric data used.

References

Peer-Timo Bremer, Herbert Edelsbrunner, Bernd Hamann, and Valerio Pascucci. A Multi-resolution Data Structure for Two-dimensional Morse-Smale Functions. In *Proceedings of IEEE Visualization 2003*, pages 139–146, 2003.

Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. The Contour Spectrum. In *Proceedings of IEEE Visualization 1997*, pages 167–173, 1997.

Roger L. Boyell and H. Ruston. Hybrid Techniques for Real-time Radar Simulation. In *Proceedings of the 1963 Fall Joint Computer Conference*, pages 445–458. IEEE, 1963.

Yi-Jen Chiang, Tobias Lenz, Xiang Lu, and Günter Rote. Simple and Output-Sensitive Construction of Contour Trees Using Monotone Paths. Technical Report ECG-TR-244300-01, Institut für Informatik, Freie Universtät Berlin, 2002.

Hamish Carr and Jack Snoeyink. Path Seeds and Flexible Isosurfaces: Using Topology for Exploratory Visualization. In *Proceedings of Eurographics Visualization Symposium 2003*, pages 49–58, 285, 2003.

Hamish Carr, Jack Snoeyink, and Ulrike Axen. Computing Contour Trees in All Dimensions. *Computational Geometry: Theory and Applications*, 24(2):75–94, 2003.

Hamish Carr, Jack Snoeyink, and Michiel van de Panne. Progressive Topological Simplification Using Contour Trees and Local Spatial Measures. Accepted to IEEE Visualization 2004.

H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Comput. Geom.*, 30:87–107, 2003.

H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Comput. Geom.*, 28:511–533, 2002.

M. Hilaga, Yoshihisa Shinagawa, T. Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *SIGGRAPH 2001*, pages 203–212, 2001.

Valerio Pascucci and K. Cole-McLaughlin. Efficient Computation of the Topology of Level Sets. In *Proceedings of IEEE Visualization 2002*, pages 187–194, 2002.

Georges Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Comptes Rendus de l'Acadèmie des Sciences de Paris*, 222:847–849, 1946.

Shigeo Takahashi, Gregory M. Nielson, Yuriko Takeshima, and Issei Fujishiro. Topological Volume Skeletonization Using Adaptive Tetrahedralization. In *Geometric Modelling and Processing 2004*, 2004.

Marc van Kreveld, René van Oostrum, Chandrajit L. Bajaj, Valerio Pascucci, and Daniel R. Schikore. Contour Trees and Small Seed Sets for Isosurface Traversal. In *Proceedings of the 13th ACM Symposium on Computational Geometry*, pages 212–220, 1997.