Topology-controlled Volume Rendering

Scott Dillard, sedillard@ucdavis.edu Gunther H. Weber, ghweber@ucdavis.edu
Hamish Carr, hamish.carr@ucd.ie Valerio Pascucci, pascucci@llnl.gov
Bernd Hamann, bhamann@ucdavis.edu

ABSTRACT

Visualization based on the topology of scalar fields defined over three-dimensional (3D) domains can provide valuable insights. Existing methods use topology to identify interesting features or to treat individual contours (connected components of isosurfaces) as distinct entities for isosurfacing or interval volume rendering. We extend these ideas to direct volume rendering. We use the contour tree to define a segmentation of the 3D domain and then apply independent color maps to topologically uniform regions of the field, hiding or highlighting regions as needed.

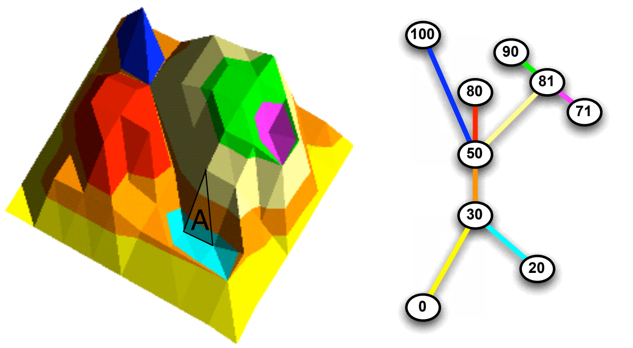
1. INTRODUCTION

Volume rendering is one of the core techniques used for scientific and medical visualization, assigning optical properties to points in the 3D domain of a scalar field and computing the resulting intensity at an image plane [8]. The color map, or *transfer function* that assigns the optical properties is applied uniformly in the field, and is normally based on the density value at a given point. While this approach provides an overview of an entire data set, it lacks the flexibility to distinguish between distinct features that share the same value. If a region of interest is enclosed by an "uninteresting" region, occlusion generally prevents the effective visualization of the interesting region.

There has been much work on defining transfer functions which express interesting features in the data. Improved methods utilize additional derived quantities such as gradient magnitude [7], but they still apply the same transfer function uniformly throughout the domain. Other methods use segmentation information to modify a transfer function [12] or to apply different rendering methods to different regions [5]. Recently, statistical learning-based methods have been introduced for interactive volume data segmentation, see [13], for example. The idea underlying such methods is to have a user specify interactively what regions in a data set constitute a "feature." By pointing out such regions, it is then possible to characterize them by scalar field behavior in a local neighborhood, and to use the resulting characterization for segmentation.

Our method is similar to segmentation-based approaches, as we use different transfer functions for different regions of the data. Our segmentation comes directly from the underlying topological structure of the data, as it is expressed by the *Reeb graph* of the scalar field.

Reeb graphs depict the level-set topology of a function. In a 2D scalar field, such as a terrain, level-sets are lines of constant elevation, like the contours on a hiker's topographic map. In 3D, they form surfaces, or *isosurfaces*. At a given elevation, the level-set will contain distinctly connected components, or *contours*. If we trace the connectivity of these contours across all elevations, we obtain a Reeb graph. The nodes of the graph represent *critical*



(a) Terrain showing topological zones (b) Contour tree

Figure 1: 2D example of a segmentation defined by a contour tree. (a) Terrain data set showing topological zone segmentation. (b) Contour tree of the terrain, with edges color-coded based on the corresponding topological zones.

points where components split or merge, such as when two peaks meet at a saddle. If the domain of the scalar field contains holes, the Reeb graph may contain cycles. Otherwise, the graph is called a *contour tree* [1]. Because scientific data sets often lack holes, and contour trees are easier to compute than general Reeb graphs, they have been given special attention in scientific visualization literature [11, 6, 14, 3, 9].

This work extends previous contour-tree-based methods for isosurface extraction. Carr and Snoeyink [2] treat the individual contours of an isosurface as separate entities. This allows for the outer component of a surface to be removed so that inner details are revealed.

2. METHOD

We generalize the method of Carr and Snoeyink in the framework of volume rendering. Whereas they extract isosurfaces corresponding to points on the edges of the contour tree, we define transfer functions along entire edges of the tree. For a given point p with density $f(p)$, we first determine which contour of that level-set contains p. This contour is represented by an edge in the tree, and the transfer function for this edge is used to color p. The functionality of "hiding" regions which occlude features of interest can be duplicated by using transfer functions which are 100% transparent for these occluding regions. Takahashi et al. [10] describe transfer functions which express topological features. Our method allows these tools to be applied pricely on a per-contour basis, rather than globally across all contours.

Often, data acquired through scanning devices contains noise which creates an unmanageable number of critical points in the contour tree. It becomes necessary then to simplify the contour tree. Edges of the contour tree which represent tiny volumes, or span small density ranges, are removed. Transfer functions are specified on edges of the simplified tree, and are then propagated to the multitude of edges which were removed. The simplified tree is stored as a hierarchical structure, so that fine details can be revealed on demand. See [4] for details on simplification of contour trees.

We have implemented our rendering algorithm on graphics hardware, running at interactive frame rates. This allows for transfer functions to be modified quickly so that data can be explored by "peeling" through the layers where topological changes occur. Hidden peaks or voids in the data can be discovered by removing outer layers and working inwards.

3. RESULTS

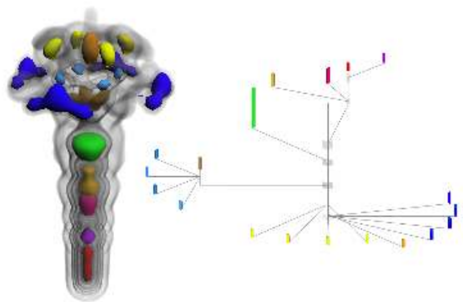


Figure 2: Simulation data set with contour tree.

Figure 3: CT scan. Note that the air in the lung is the same density as the air outside the body, yet rendered differently. The other lung was removed to show the ribs.

Acknowledgments

This work was supported by the National Science Foundation under contracts 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 a large Information Technology Research (ITR) grant; the National Institutes of Health under contract P20 MH60975-06A2, funded by the National Institute of Mental Health and the National Science Foundation; the National Institutes of Health under contract 1R01 GM70444-01A1, funded by the National Institute of General Medical Science; by the Director, Office of Science, U.S. Department of Energy under contract DE-AC03-76SF00098; and the Lawrence Berkeley National Laboratory (LBNL).

4. REFERENCES

[1] R. 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.

[2] H. Carr and J. Snoeyink. Path seeds and flexible isosurfaces using topology for exploratory visualization. In *Data Visualization 2003 (Proceedings VisSym 2003)*, pages 49–58, New York, NY, 2003. ACM Press.

[3] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry – Theory and Applications*, 24(2):75–94, Feb. 2003.

[4] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *IEEE Visualization 2004*, pages 497–504. IEEE, Oct. 2004.

[5] H. Hauser, L. Mroz, G.-I. Bischi, and M. E. Gröller. Two-level volume rendering - fusing mip and dvr. In *IEEE Visualization 2000*, pages 211–218, Los Alamitos, California, 2000. IEEE Computer Society Press.

[6] T. Itoh and K. Koyamada. Automatic isosurface propagation using an extrema graph and sorted boundary cell lists. *IEEE Transactions on Visualization and Computer Graphics*, 1(4):319–327, 1995.

[7] J. Kniss, G. Kindlmann, and C. Hansen. Multidimensional transfer functions for interactive volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 8(3):270–285, 2002.

[8] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, May 1988.

[9] V. Pascucci and K. Cole-McLaughlin. Parallel computation of the topology of level sets. *Algorithmica*, 38(2):249–268, Oct. 2003.

[10] S. Takahashi, I. Fujishiro, and Y. Takeshima. Interval volume decomposer: A topological approach to volume traversal. In R. F. Erbacher, K. Börner, M. Gröhn, and J. C. Roberts, editors, *Visualization and Data Analysis 2005 (Proceedings of the SPIE)*, 2005.

[11] S. Takahashi, T. Ikeda, Y. Shinagawa, T. L. Kunii, and M. Ueda. Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data. *Computer Graphics Forum*, 14(3):181–192, 1995.

[12] U. Tiede, T. Schiemann, and K. H. Höhne. High quality rendering of attributed volume data. In *IEEE Visualization 1998*, pages 255–262, Los Alamitos, California, 1998. IEEE Computer Society Press.

[13] F.-Y. Tzeng, E. Lum, and K.-L. Ma. An intelligent system approach to higher-dimensional classification of volume data. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):273–284, May/June 2005.

[14] M. van Kreveld, R. van Oostrum, C. Bajaj, V. Pascucci, and D. R. Schikore. Contour trees and small seed sets for isosurface traversal. In *Proceedings of the 13th ACM Annual Symposium on Computational Geometry (SoCG)*, pages 212–220. ACM Press, 1997.