Scalar Field Visualization: Level set topology

Vijay Natarajan
Department of Computer Science and Automation
Supercomputer Education and Research Centre
Indian Institute of Science, Bangalore

1 Scalar Functions

A function uniquely maps members of one set to members of another set. We are interested in functions that map points from a n-dimensional space to real values. These functions are called scalar functions. Scientific datasets are often representations of functions but typically do not have an analytic description of the function. In fact, the value of a function is measured at discrete points in the domain by physical means or computed using procedural methods. Different interpolation techniques are then used to derive a continuous and, if necessary, smooth representation of the function. For example, X-ray crystallographers compute the electron density at various points of a molecular crystal using diffraction measurements from x-rays bouncing on the crystal. It is essential to know the electron density to perform structure related studies of the molecule. The electron density is a scalar function typically defined on a subset of the three-dimensional Euclidean space, $\mathbb{R}^3$. Another example is Magnetic Resonance Imaging (MRI), a popular technique used to take pictures of different slices of the human body. The atom density over the slice is mapped to a gray-scale image and studied by radiologists to detect tumors. The scalar function in this case is the density defined on a set of two-dimensional planes stacked together and can be viewed as a function defined on a subset of $\mathbb{R}^3$.

2 Visualization

Humans are able to easily interpret and comprehend visual information. The field of data visualization capitalizes on this ability and aims to give the user a deeper understanding of the data and underlying laws governing it. This is achieved by providing a comprehensive display of the data along with annotations. A simple visualization method interprets the scalar value at each point as an additional coordinate and plots a graph. Obviously, graph plots are useful only if the domain is one- or two-dimensional. Other techniques like isosurface extraction and volume rendering are used to visualize three-dimensional scalar data. The book by Hansen and Jonson [5] provides a good introduction to the field of visualization.

A severe limitation of many scalar field visualization techniques is the inability to handle huge and feature-rich datasets, which are becoming increasingly common. One approach towards solving this problem is to compress the data using geometric techniques and visualize the compressed data. Another approach is to automatically extract features from the dataset and present them to the user. Computing the topological properties of the domain is a step in the latter direction and has been studied under different names [1, 4].

3 Level Sets

A smooth, real-valued function $f : \mathbb{M}^d \rightarrow \mathbb{R}$ defined on a d-manifold is called a *Morse function* if it satisfies the following conditions:

1. all critical points of f are non-degenerate and lie in the interior of $\mathbb{M}^d$,
2. all critical points of the restriction of f to the boundary of $\mathbb{M}^d$ are nondegenerate, and
3. for all pairs (p, q) of distinct critical points of f and its restriction to the boundary, $f(p) \neq f(q)$.

In the following discussion, we assume that the given function is Morse. The above conditions may not hold in practice. However, the input can be suitably preprocessed to ensure that the assumptions hold.

The preimage of a real value is called a *level set*. The level set of a Morse function f is a $(d-1)$-manifold with or without boundary, possibly containing multiple connected components. For the case when $d = 3$, a level set is called

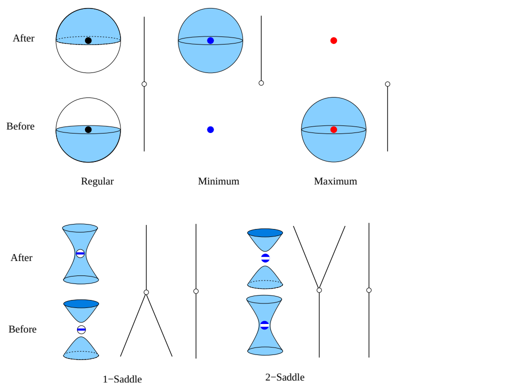

Figure 1: Figures above show the isosurfaces before ($f^{-1}(c-\epsilon)$) and after ($f^{-1}(c+\epsilon)$) passing through a point with function value c and the structure of the Reeb graph at the corresponding node. Topology of the isosurface changes when it evolves past a critical point.

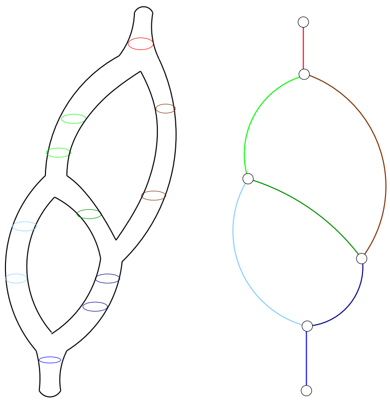

Figure 2: Reeb graph of the height function defined on a surface with two tunnels. Reeb graph tracks the topology of level sets.

an *isosurface*. We are interested in the evolution of the isosurface as the function value increases. Significant changes occur at critical points. Specifically, the topology (*i.e.*, connectivity) of the isosurface changes either by gaining / losing a component or by gaining/losing genus. No topological changes occur at regular points. Figure 1 illustrates the various topology changes that occur at critical points. The isosurface gains a component when it evolves past a minimum and loses a component when it evolves past a maximum. At 2-saddles, the local pictures in Figure 1 indicate an apparent splitting of a component into two. Global behavior of the isosurface component will determine if this is indeed a split or a reduction in genus.

4 Reeb Graphs

The *Reeb graph* of f is obtained by contracting each connected component of a level set to a point, see Figure 2. The Reeb graph expresses the evolution of connected components of level sets as a graph whose nodes correspond to critical points of the function. Figure 1 illustrates the structure of the Reeb graph at various types of nodes. In the case of saddles, the corresponding node has degree 3 if the saddle merges/splits components, and degree 2 if it is a genus modifying saddle. The Reeb graph tracks topology changes in level sets of a scalar function and finds applications in scientific visualization and geometric modeling. They serve as an effective user interface for selecting meaningful level sets and transfer functions for volume rendering.

We describe two algorithms that construct the Reeb graph of a Morse function defined on a 3-manifold. The first algorithm [3] maintains the connected components of levels sets as a tree-cotree decomposition and constructs the Reeb graph in $O(n \log n + n \log g (\log \log g)^3)$ time, where n is the number of triangles in the tetrahedral mesh representing the 3-manifold and g is the maximum genus over all level sets of the function. We also extend this algorithm to construct Reeb graphs of d-manifolds in $O(n \log n (\log \log n)^3)$ time, where n is the number of triangles in the simplicial complex representing the d-manifold. Our result is a significant improvement over the previously known $O(n^2)$ algorithm. Experimental results indicate that, in practice, our algorithm for 3-manifolds performs better than what the theoretical bound suggests.

The second algorithm [2] is near-optimal and has the added advantage that it is simple to implement. This algorithm first identifies the critical points of the input, and then connects these critical points in the second step to obtain the Reeb graph. A simplification mechanism based on topological persistence aids in the removal of noise and unimportant features.

References

[1] C.L. Bajaj, V. Pascucci, and D. Schikore. Visualization of scalar topology for structural enhancement. In *Proc. IEEE Conf. Visualization*, pages 51–58, 1996.

[2] H. Doraiswamy and V. Natarajan. Efficient output-sensitive construction of Reeb graphs. In *Proc. Intl. Symp. Algorithms and Computation*, 2008.

[3] H. Doraiswamy and V. Natarajan. Efficient algorithms for computing Reeb graphs. *Computational Geometry: Theory and Applications*, 2009, to appear.

[4] A. T. Fomenko and T. L. Kunii. *Topological Modeling for Visualization*. Springer Verlag, 1997.

[5] C. D. Hansen and C. R. Johnson, editors. *Visualization Handbook*. Academic Press, 2004.