Contour Tree and Morse-Smale - Option 2

Yihong Liu*
Oregon State University

Ziyu Xiong*
Oregon State University

Abstract

Our idea on this final term project is based on the research paper "Visualizing 2D Scalar Fields with Hierarchical Topology" [Wu and Zhang 2015]. This article describes the work of creating new visualizations by using hierarchical scalar topology. We are mainly introduce Contour Tree (CT) and Morse-Smale (MS) complex scalar fields to visualize our dataset. There is another method called Topology Simplification which we will not go through details in this proposal, but it is definitely a useful tool in many other cases. Contour Tree is the main goal to achieve in this project and we will also have some discuss and attempts on Morse-Smale complex scalar fields.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems;

Keywords: scalar topology, contour tree, morse-smale complex, multivariate map, contouring, spaghetti plot

1 Introduction

Scalar Topology is a very useful tool on many fields of scientific visualization, such as remeshing the surfaces and using as shape descriptor.
Firstly, contour trees were mainly used to compute the seed sets, to trace partial or whole isosurfaces and determine important values of the height function where topological changes occur in the level sets; these changes may correspond to important phenomena in the data studied[Carr et al. 2003]. Contour Tree describes the nesting relation of contours through a tree structure with each arc representing a uniform contour component and each node describing a topological change of the component at a critical point.[Wu and Zhang 2015]
Secondly, The most complete description of the topology of a scalar field is its Morse-Smale complex which segments the field based on its gradient. While the usefulness of the Morse-Smale complex has been widely acknowledged it has rarely been applied in practice[Bremer and Pascucci 2007]. A MS complex decomposes a scalar field into quadrangular cells with uniform gradient flow behavior[Wu and Zhang 2015].

2 Previous Work

In class, we have already talked about the basic concepts of contour lines and how to draw contour lines in mesh, as well as the ways of finding critical points, especially in quad mesh.

*e-mail: {liuyi|xiongz}@eecs.oregonstate.edu

2.1 Contour Line

Firstly, we need to loop through all the scalar values at each point to check how many points are in the target contour. Secondly, we need to loop through the vertices for each margin square. If there is one point in the object and another is not, then mark the middle point for future use. Lastly, loop through all the quad in the mash and connect those previously marked middle points.

2.2 Critical Point

In order to classify the critical point, it is very important to know the concepts of piecewise quadratic, as well as how to apply bilinear interpolation in each cell. In this way, firstly, we need to go through all the points in the mesh and find out the points that

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial y} = 0$$

Secondly, in each cell, we assign the four points as (x_1, y_1), (x_1, y_2), (x_2, y_1) and (x_2, y_2). By calculating the partial derivative of

$$f(x,y) = \frac{x_2 - x}{x_2 - x_1}\frac{y_2 - y}{y_2 - y_1}f(x_1,y_1) + \frac{x - x_1}{x_2 - x_1}\frac{y_2 - y}{y_2 - y_1}f(x_2,y_1) + \frac{x_2 - x}{x_2 - x_1}\frac{y - y_1}{y_2 - y_1}f(x_1,y_2) + \frac{x - x_1}{x_2 - x_1}\frac{y_2 - y}{y - y_1}f(x_2,y_2)$$

It is not difficult to find out critical points that

$$x_0 = \frac{y_2 f(x_1,y_1) - y_2 f(x_2,y_1) - y_1 f(x_1,y_2) + y_1 f(x_2,y_2)}{f(x_1,y_1) - f(x_2,y_1) - f(x_1,y_2) + f(x_2,y_2)}$$

$$y_0 = \frac{x_2 f(x_1,y_1) - x_1 f(x_2,y_1) - x_2 f(x_1,y_2) + x_1 f(x_2,y_2)}{f(x_1,y_1) - f(x_2,y_1) - f(x_1,y_2) + f(x_2,y_2)}$$

To determine the classification of critical points, we need to calculate the second partial derivative of $f(x, y)$. Using those partial derivative to form a Hessian matrix, if eigenvalues of Hessian matrix are both negative, then this critical point is local maximum. If eigenvalues of Hessian matrix have both positive and negative values, then it is a saddle point [Zhang 2019].

3 Background

3.1 Contour Tree

A graph-based representation to illustrate how the topology of level set changes with the scalar values. For a given value h , the level set of f at h is the subset $L(h) = \{x \in M | f(x) = h\}$. We call each connected component of level set $L(h)$ a contour. As h increases in the level set of $L(h)$, contours appear at local minima, join or split at saddles, and disappear at local maxima of f . With each contour represented as a node, the Contour Tree for a Morse function is a graph that tracks the evolution of contours [Wu and Zhang 2015].

3.2 Morse-Smale

Morse-Smale is aiming at investigating the topology of a surface by looking at critical points of a function on surface. In addition, all the critical points are isolated and non degenerate

$$\nabla f(x) = \begin{pmatrix} \frac{\partial f}{\partial p} f(p) & \frac{\partial f}{\partial y} f(p) \end{pmatrix} = 0$$

$$det(Hessian(p)) \neq 0$$

A Morse-Smale Complex can be obtained by superimposing the descending and ascending manifolds of f. Within a MS complex cell, all integral lines start at the same minimum and end at the same maximum.

3.3

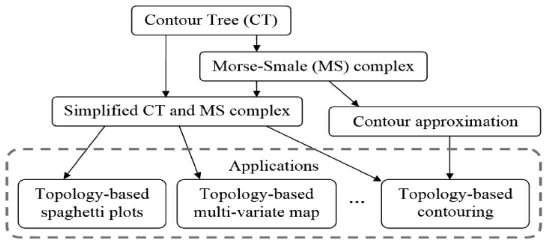

Figure 1: The visualization framework based on hierarchical topology

4 Result

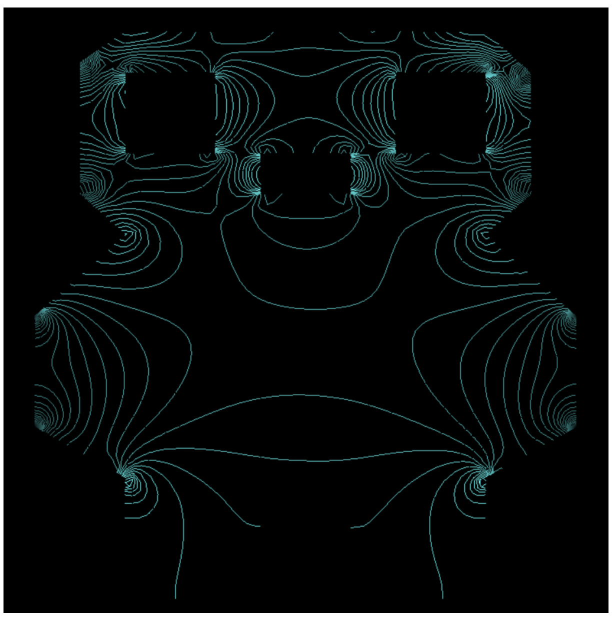

Figure 2: Results of minecraft ply file using Contour line

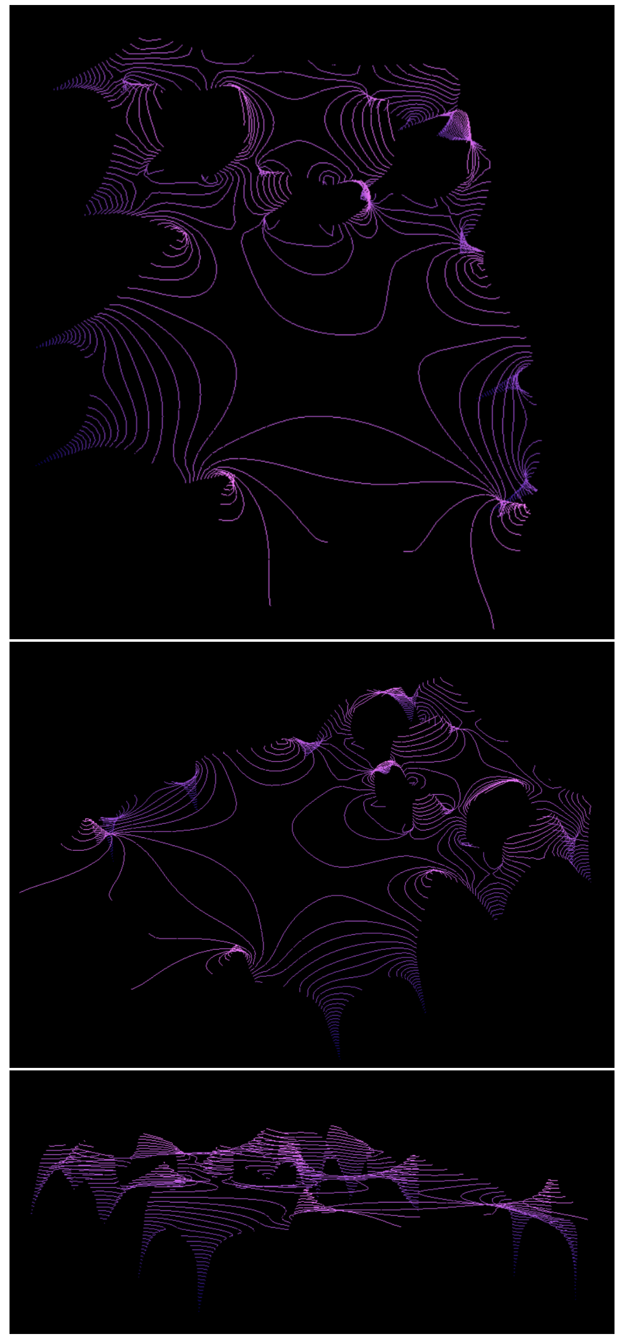

Figure 3: Results of minecraft ply file using contour line with height

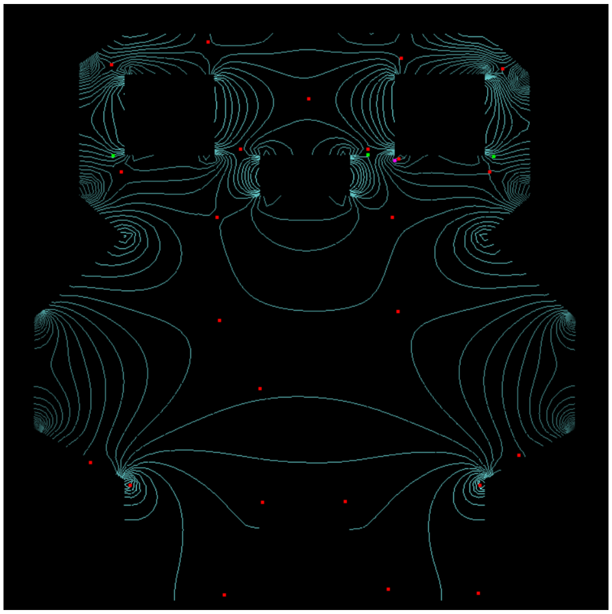

Figure 4: Results of minecraft ply file using contour line and critical points

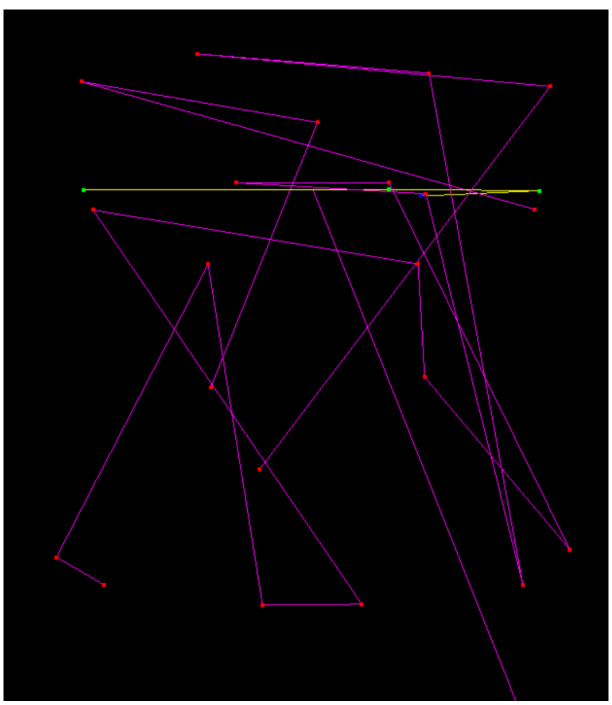

Figure 5: Results of minecraft ply file using contour line and critical points

5 Evaluation

5.1 Contour Line

The result of **Contour Line** is successfully showing us the contours of different level sets. This method is very basic and straightforward, which also provides the basis of the following methods.

5.2 Contour Line with Height and Critical Points

In this method, we have put multiple results from different angles, which shows us how the critical points distributed stereoscopically. Combine the height method with the critical points method, We can roughly verify the accuracy of these critical points. It is worth mentioning that there are lots of boundary points with a "local maximum" or "local minimum" feature. However, theoretically, we still cannot pretend these points as actual critical points. We will talk more about this in following method of connecting critical points.

5.3 Contour tree - Lines Between Critical Points

In this method, we mainly work on connecting all the critical points in the mesh. In the contour tree algorithm that I choose for final project, when we reach a critical point that is local maximum, we start a new component and activate the cells next to the local maximum. When we reach a critical point that is local minimum, we destroy the component and this point is set as the end leaf node. The saddles are the points that determine merging or splitting the component. There are 28 critical points in total, including 24 saddle points, 3 local maximum and 1 local minimum. The issue in this project is the unbalanced critical points. There are too many saddle points, with only 3 local maximum and 1 local minimum. In the algorithm, local maximum and minimum are 2 main factors grow and end the tree. However, the lack of local maximum and minimum will directly limit the contour tree. As we can see the yellow lines in figure 5, it connects 3 local maximum (green dots) and ends at local minimum (blue dot). The local minimum really limits the contour tree. Meanwhile, an excess of saddle points really bring us so many obstacles to handle. We did not figure out a perfect way to deal with these saddle points, we just connect these saddles one by another. We are looking forward to see more useful methods to improve the contour tree [Han-Wei'Shen 2010].

6 Division of tasks

For the contour tree project, there are two main difficult points, the first one is to find all the "important" points on the image, such as, saddle point, local min points, local max points, the second one is to connect the points with line, how are we going to decide the algorithm to connect different types of points. Yihong Liu did the most portion for part one, which is to find all the points that we need to draw the contour tree, Ziyu Xiong did the most portion for part two, which is developing the algorithm to connect those points. We also helped each other to do their part's work, not just splitting the whole project into two projects, because we meed to make sure the the way we find or sort points can be used in the "connecting" algorithm and make the contour tree work.

7 Conclusion

To sum up, we really have a struggling time in the implementation of final project. Firstly, we want to appreciate all the helps and suggestions given by Professor Zhang and Steve (Zixuan). We gain the ideas from the example code provided by Steve, authored by Shiyao. In addition, we are very thankful for another group who also implements topic in contour tree. We had conversations and deep communications in the library, which really opened our minds to implement our method. From a technical perspective, we are still not doing perfect on the contour tree algorithm, such as how to correctly deal with limit number of local maximum and local minimum values. Also, we did not figure out the right way to merge and split contour trees. Although part of the issues can be caused by the special dataset we have for this project, but we still have lots of space to improve our method.

References

AMIT CHATTOPADHYAY, HAMISH CARR, D. D. Z. G. O. S. 2014. Multivariate topology simplification.

BREMER, P.-T., AND PASCUCCI, V. 2007. A practical approach to two-dimensional scalar topology.

CARR, H., SNOEYINK, J., AND AXEN, U. 2003. Computing contour trees in all dimensions. *Computational Geometry 31* (Feb.).

HAN-WEI'SHEN. 2010. Contour trees. *Ohio State University*.

WU, K., AND ZHANG, S. 2015. Visualizing 2d scalar fields with hierarchical topology. *2015 IEEE Pacific Visualization Symposium (PacificVis) 16* (July).

ZHANG, E. 2019. Scalar field topology. *CS 453/553 Scientific Visualization*.