LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-670379

Rendering the Topological Spines

D. Nieves-Rivera

May 5, 2015

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Rendering the Topological Spines

Delma I. Nieves-Rivera

Office of Science, Science Undergraduate Laboratory Internship (SULI)

Center for Applied Scientific Computing (CASC)

Lawrence Livermore National Laboratory

Livermore, California

Spring 2015

Disclaimer

Table of Contents

ABSTRACT

Rendering the Topological Spines. DELMA I. NIEVES-RIVERA (University of Puerto Rico, Arecibo, PR, 00614), TIMO BREMER (Lawrence Livermore National Laboratory, Livermore, CA, 94550)

Many tools to analyze and represent high dimensional data already exits yet most of them are not flexible, informative and intuitive enough to help the scientists make the corresponding analysis and predictions, understand the structure and complexity of scientific data, get a complete picture of it and explore a greater number of hypotheses. With this in mind, N-Dimensional Data Analysis and Visualization (ND^2AV) is being developed to serve as an interactive visual analysis platform with the purpose of coupling together a number of these existing tools that range from statistics, machine learning, and data mining, with new techniques, in particular with new visualization approaches. My task is to create the rendering and implementation of a new concept called topological spines in order to extend ND^2AV's scope. Other existing visualization tools create a representation preserving either the topological properties or the structural (geometric) ones because it is challenging to preserve them both simultaneously. Overcoming such challenge by creating a balance in between them, the topological spines are introduced as a new approach that aims to preserve them both. Its render using OpenGL and C++ and is currently being tested to further on be implemented on ND^2AV. In this paper I will present what are the Topological Spines and how they are rendered.

I. INTRODUCTION

While tools to analyze and represent high dimensional data already exist, many of the advanced and flexible ones are not easily accessible to scientists. They often use scatter plots and the intuition of trained experts to make analysis and predictions. These tools are not wrong but they don't provide the opportunity to explore a greater number of hypotheses nor analyze the data in a better way. In order to help scientists understand the structure and complexity of scientific data, different visualization tools were developed to provide images that are very informative and intuitive. Several of these techniques were incorporated into N-Dimensional Data Analysis and Visualization (ND^2AV). In order to broaden the scope of ND^2AV, since its still being developed, more visualization techniques needs to be added into it. A new concept that was developed, called Topological Spines, is the next target to be added. The purpose of this project is to create the corresponding rendering for it.

A. Terminology

We will be using some topology terms since is the subject in which we are working on. In this section I will define them for a better understanding of the paper. We have topology, which is the mathematical study of how the geometric objects intrinsically connect to themselves[1]. Then we talk about the maxima and minima, which is when a function decreases or increases in all directions respectively. Additionally we have the extrema, which is the union of the maxima and the minima together. In other words is the largest and smallest value the function takes at a point. Moreover we have the saddle which is when a function switches between decreasing and increasing 4 times around a point[2]. Also we have the critical point, which is when the derivative of any value in the domain of a differentiable function (a function that has a derivative for each of its point in the domain)[3] is zero[4]. For our case, some examples of critical points are a saddle, a maxima or a minima. Then we have a critical contour, which is a contour that passes right through the saddle.

It will be also mentioned a scalar field. This is a function that gives a single value of a variable for every point in space[5]. And lastly, we talk about a force directed layout algorithm, which the one used to calculate the layout of simple undirected graphs[6].

B. N-Dimensional Data Analysis and Visualization (ND^2AV)

ND^2AV is an interactive visual analysis platform that couples together a number of existing tools from statistics, machine learning, and data mining, with new techniques, in particular with new visualization approaches. From traditional analysis approaches like dimension reduction, clustering with neighborhood graphs and topological analysis to custom capabilities, ND^2AV provides an intuitive way for scientists to form their hypothesis out of the visualized data. Moreover, it gives a complete view of the data.

ND^2AV is a Python/C++ application that uses a Model-View-Controller paradigm (modular design) allowing it to be adaptable and extensible. It is created using simple drag and drop techniques, which enables the creation of sophisticated hierarchical workflows. The modules on the workflows are cross-linked permitting the users see the effect that any choice of parameter has in other results.[7] It has the ability to perform a large number of different analysis steps in the minimum time. Plus it provides the users the opportunity to experiment with different parameters and correlate the results.

ND^2AV it's currently functional and has been used into many applications. But, in order to broaden its scope, new visualization techniques needs to be incorporated into it. In the next section, I describe the topological spines; the latest concept that is in process to be added into it.

C. Topological Spines

Different visualization approaches like contour trees[8] and topological landscapes[9] are used nowadays for topological analysis. However, each of these focuses on representing either the topological properties or the structural (geometric) ones. Creating a visual representation that preserves them both is very challenging. On one hand, as the dimensionality of the data set increases, less information can be preserved and global properties often disappear. On the other, wanting to preserve all the structural properties makes the representation become even more difficult to visualize without having either occlusion or clutter problems. So the real challenge lies in finding a representation that can manage a balance of both: the topological and the geometric properties. This is why the Topological Spines is introduced as a new approach that aims to preserve them both. We can see how a topological spines looks like on figure 1.

The Topological Spines is, according to C. Correa et. al., "a new visual representation aimed at preserving both topological and structural (geometric) properties of a scalar field"[10] as mentioned above. The topological and structural (geometric) properties that it aims to preserve are: how the contours are merged and nested, the distribution, size and density of the function values, the relative location of extrema, and how the volume is preserved. They link together a number of critical points using canonical visual representations. The term canonical it's given to it because it abstracts the nesting and volume of the contours. Using this new concept we are able to represent high dimensional data in lower dimensions providing the opportunity to understand it in a better way.

The topological spines consist of: a maxima or minima, a saddle, different contours and function values. Each contours/function value has a radius that is proportional to the volume of the topological spine. Figure 2 identifies some parts of the topological spines.

We can think of the topological spines as a terrain. Following this metaphor, the maxima will be the hill/peak (as shown in figure 3), the minima will be the valley (as shown in figure 4) and the saddle (as shown in figure 5). Also the contours and function values will be the contour lines (as shown in figures 3-5).

II. METHODS

A. Research

For the making of this project I needed to familiarize myself with some subjects in order to be able to develop the rendering for the topological spines. To begin with, I needed to learn about visualization and topology in general. Then I started learning OpenGL and how to use it with C++. To accomplish this I read some papers, tutorials and researched online. After learning the basics, I was able to start working on the project. Even though, the whole process was a constant learning.

B. Drawing the topological spines

The topological spines are rendered using OpenGL and C++. In order to represent it, some basic input is needed. These are: the function value of the saddle and the maxima's along with their respective locations of the vertices, the contours that the user wants to see, and how many contours will be computed (n). With this information, the function proceeds to compute the information for the contours that are in between the maxima and saddle's function values. It calculate n (X, Y) position of points from the maxima's function value to the saddle's one as well as n (Z) values. The Z acts as both: the function value and the contour value. Additionally, it will calculate the radius for each of the different contours based on the full volume. All of this is done by linear interpolation. Then it proceeds to identify which of the contours the user wants to see in order to draw them.

The contours are drawn using their respective computed information. For each contour, a circle is drawn around the maxima with a size of its radius (as shown in figure 6). Then the tangent points of the circle are drawn. This is done by computing the tangent lines with respect to either the X and Y positions or the saddle depending on if the contour is above or below the saddle's contour. If the contour is above it, the spine is drawn pinched (as shown in figure 7). On the other hand, if the contour is below it, the spine is draw opened (as shown in figure 8). For drawing the spine pinched, we draw a triangle from the tangent points of its circle to the X and Y position where it pinches. For drawing the spine open, we draw a quad from the tangent points to the two points where it pinches at the saddle. An example of a full spline can be found in figure 9.

The process described above is the one used to draw one side of the topological spine consisting of a maxima and a saddle. In order to draw a full triplet (maxima-saddle-maxima), the function is called twice with the corresponding input values.

C. Force directed layout

The input data used to render the topological spines and make the corresponding tests was hard-coded. But, since we want the topological spine to be soft-coded in order to provide more flexibility, a force directed layout was implemented. Such layout presents the topological spine as a mass-spring model in order to calculate the locations of the vertices. A simple example of a force directed layout was made to familiarize myself with the concept and verify that everything was working correctly. Then, the algorithm for drawing the topological spines was incorporated into it. Using this layout, the positions of the maxima's and the saddles will be randomly computed, the draw spine function will be called, and we will be able to see n numbers of spines linked together. The layout implements Hooke and Coulomb's law to calculate the forces that will act on the nodes to help preserve the distances in between them. From figure 8 to 10 we can see how the layout works. Moreover, the user will have the ability to move around the topological spines in order to assign the vertices a new location. The user can select one of the spines and move it to the desired place. Once it's released, the model will recalculate the information in order to redisplay the model correctly.

III. RESULTS

As for results, the rendering of the topological spines along with the force directed layout is currently working. In figure 10 to 12 we can see a topological spine rendered showing all its properties. From figure 13-16 we can see the rendering of the topological spines along with the force directed graph.

IV. CONCLUSIONS

In conclusion, we obtain the rendering of the topological spines preserving all the structural and topological properties that we wanted. Because things took longer than expected to accomplish, the topological spines has not been tested with the real data yet. It is ready to be tested with and implemented into ND^2VA to finalize the process of adding a new visualization technique into it. During the making of this research I learned a lot while practicing some of my skills. It was a very enriching experience for my personal and professional development.

V. ACKNOLEDGEMENTS

I will like to thank the Department of Energy (DOE) for giving me the chance to participate in the Science Undergraduate Laboratory Internship (SULI) and the Lawrence Livermore National Laboratory (LLNL) for hosting me for the past months. To Barry Goldman for coordinating the program and always being willing to help; And to Timo Bremer for mentoring and giving me the opportunity to participate in his research project. Thank you.

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-TR-670379

VI. REFERENCES

[1] Topology. Retrieved from: http://web.ornl.gov/sci/ortep/topology/defs.txt

[2] P.-T. Bremer, B. Hamann, H. Edelsbrunner, V. Pascucci. "A topological hierarchy for functions on triangulated surfaces". IEEE Transactions on Visualization and Computer Graphics, 10 (4), 1077-2626 (2004)

[3]Differentiable function. Retrieved from: http://mathworld.wolfram.com/Differentiable.html

[4]Critical point. Retrieved from: http://mathworld.wolfram.com/CriticalPoint.html-

[5]Scalar field. Retrieved from: http://web.mit.edu/viz/EM/visualizations/coursenotes/modules/guide01.pdf

[6] Force directed graph. Retrieved from: http://arxiv.org/pdf/1201.3011v1.pdf

[7] P.-T. Bremer, D. Maljovec, A. Saha, B. Wang, J. Gaffney, B. Spears, V. Pascucci. "ND^2AV: N-Dimensional Data Analysis and Visualization, Analysis for the National Ignition Campaign"

[8] Contour trees. H. Carr, J. Snoeyink, and U. Axen. "Computing contour trees in all dimensions." Computational Geometry, 24 (2), 75–94 (2003)

[9] Topological landscapes. G. Weber, P.-T. Bremer, and V. Pascucci. "Topological landscapes: A terrain metaphor for scientific data". IEEE Transactions on Visualization and Computer Graphics, 13 (6), 1077–2626 (2007).

[10] C. Correa, P. Lindstrom, and P.-T. Bremer. "Topological Spines: A Structure-preserving Visual Representation of Scalar Fields". IEEE Transactions on Visualization and Computer Graphics, 17 (12), 1842-1851 (2011)

[11]Retrieved from: http://www.offroadsafety.org/get-out/land-navigation/terrain-features-on-a-map-latlon

[12] P.-T. Bremer, B. Hamann, H. Edelsbrunner, V. Pascucci. "A topological hierarchy for functions on triangulated surfaces". IEEE Transactions on Visualization and Computer Graphics, 10 (4), 1077-2626 (2004)

VII. APPPENDIX

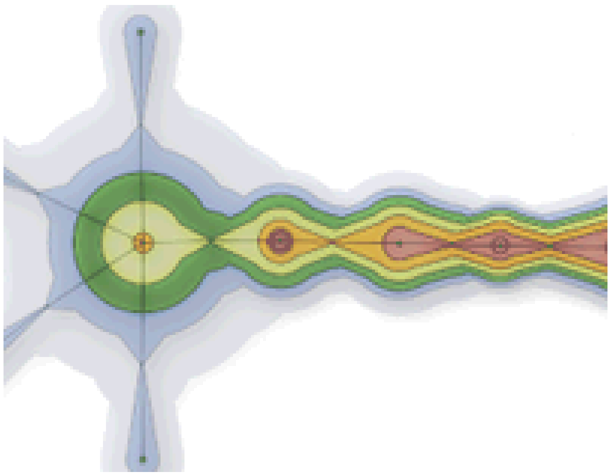

Figure 1(Courtesy from [12])

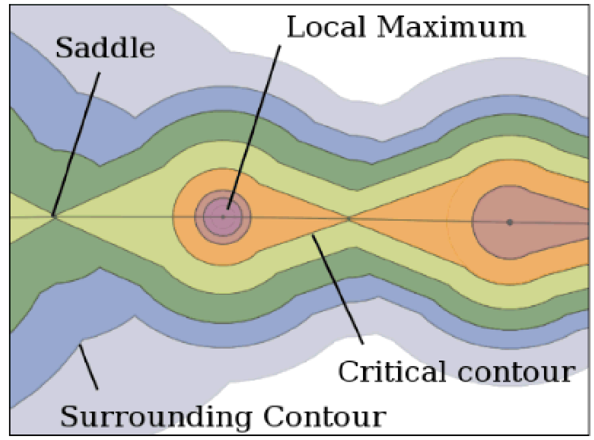

Figure 2 (Courtesy from [12])

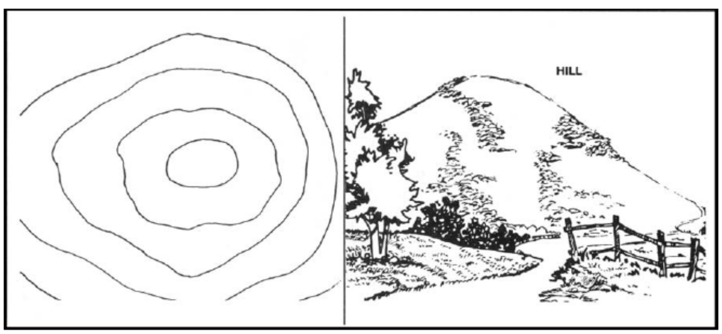

Figure 3 (Courtesy from [11])

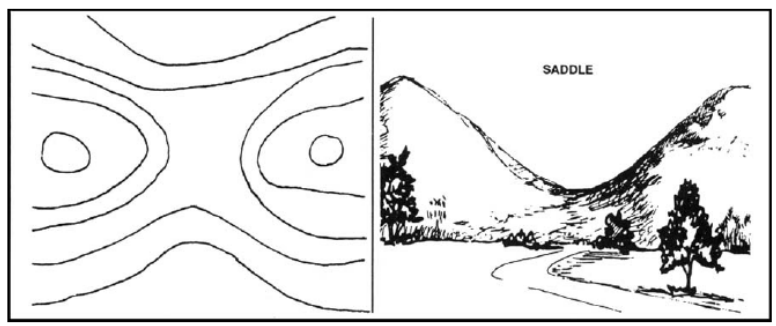

Figure 4 (Courtesy from [11])

Figure 5 (Courtesy from [11])

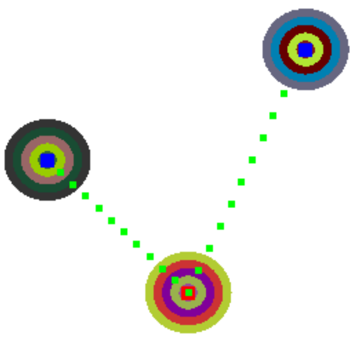

Figure 6

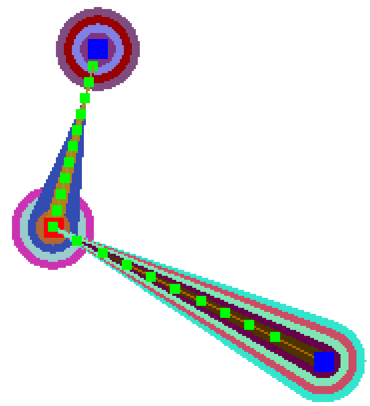

Figure 7

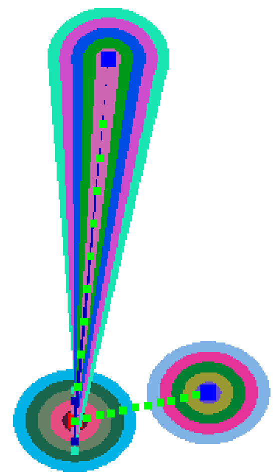

Figure 8

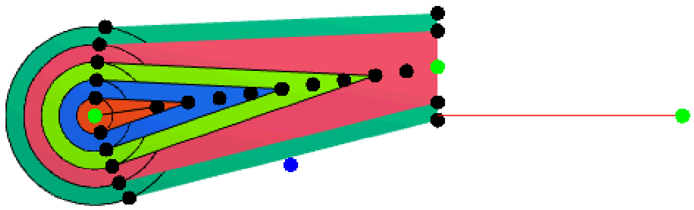

Figure 9

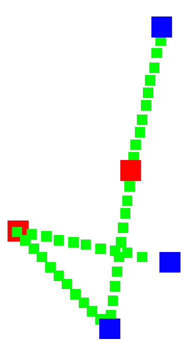

Figure 10

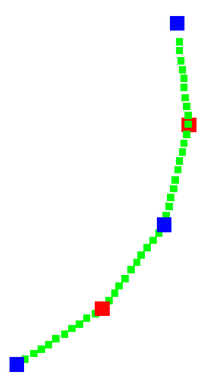

Figure 11

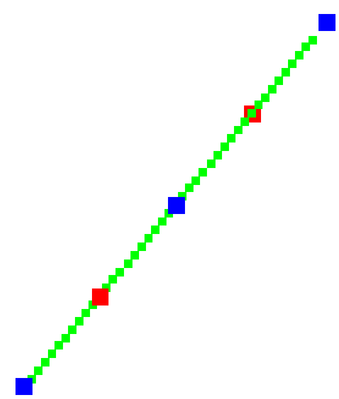

Figure 12

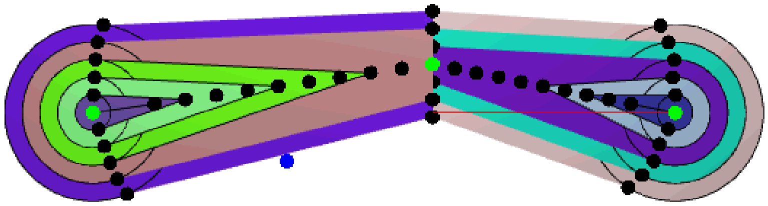

Figure 13

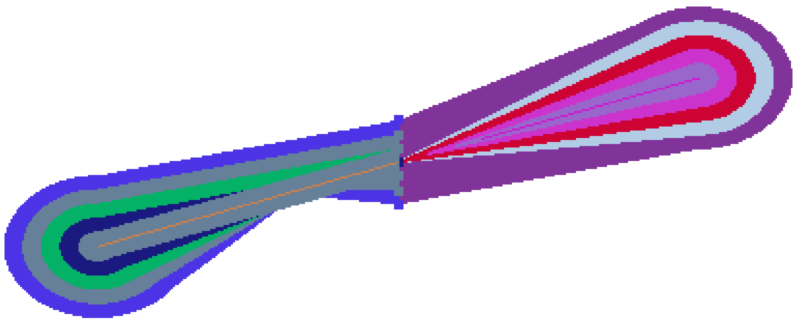

Figure 14

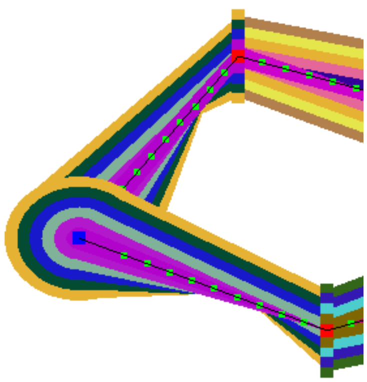

Figure 15

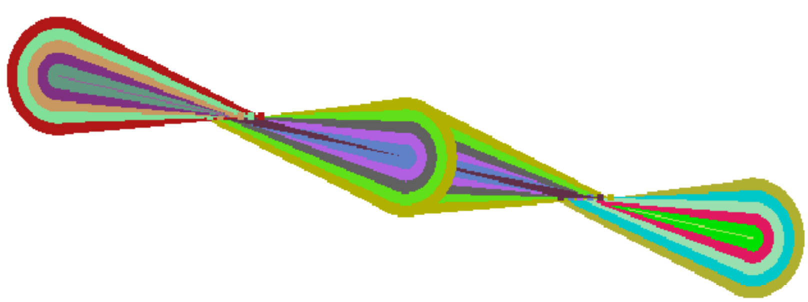

Figure 16