

博士論文

Extracting and Visualizing Singular Fibers for  
the Analysis of Multivariate Data

(多変量データ解析のための  
特異ファイバー抽出と可視化)

櫻井 大督  
Daisuke Sakurai

## Abstract

Scalar topology has provided computational tools that analyze and visualize data for scientific and engineering tasks. Scalar topology became popular in visualization, since exploring the topology of isocontours allows us to understand the singularities, which are characterized as the topological transitions of isocontours. In case of terrain datasets, for instance, a loop-shaped isocontour shrinks to a point at the local maxima of height values. For multivariate data, on the other hand, isocontours generalize to fibers – inverse images of multivariate function values – and therefore the generalized topology is known as fiber topology.

In contrast to the maturity of the single scalar case, establishing computational techniques for multivariate topology has long faced challenges. Especially, the number of degrees of freedom for tracking a fiber is greater than one. Indeed, existing techniques for the single scalar case are not directly extensible since they relied on the uniqueness of the number of data values. In fact, tracking a fiber has been accomplished by sorting the sampling points with respect to their scalar value. Unfortunately, this strategy is not directly applicable to the fiber tracking for multiple degrees of freedom.

Therefore, this thesis aims at the method of extracting singular fibers at which topological changes occur, and the applications of the method to visualizing multivariate data. To find the singular fibers, tracking the fibers with multiple degrees of freedom is necessary. To overcome this problem, the data values are quantized. The target is multivariate data in the form of  $\mathbf{R}^3 \rightarrow \mathbf{R}^2$ , where the domain is a 3D sampling space and the range consists of two variables at the 3D samples. The advantages of studying this form are that it is the simplest situation with non-trivial topology, and that their singular fibers are well studied. Furthermore, it can be extended to higher dimensional cases by accumulating specific knowledge for handling multiple variables. The central idea of this work is to track the topological changes of fibers and identify the associated topological types on the basis of singular fiber theory. The advantage of this approach is that one can understand singularity of fibers, with their connectivity across different function

values. This information benefits mathematicians for understanding multivariate functions, and data analysts for selecting a handful of features from an overwhelming amount of them.

The contributions are threefold. Firstly, the singular fibers are extracted and their topological shapes are identified. This permits the characterization of the topological changes in the fibers. To do so, connectivities between fibers are extracted in the interior of data and on the boundary, and these two types of connectivities are investigated for identifying the detailed types of singularities. Unfortunately, fibers in practical datasets can have a large number of critical points in fibers. This fact makes it difficult to directly employ the original classification in the mathematical theory of singular fibers. To deal with such non-ideal behavior, we firstly identify topology transitions of fibers such as birth and split, and then further identify the topological shapes of singular fibers if they meet the theoretical classification in mathematics.

Secondly, the proposed approach is applied to the mathematical study of the fiber topology in analytic functions. A visualization interface for domain experts is demonstrated. When exploring fiber topology, domain experts have relied on manual visualization, which does not fully facilitate them to understand complicated topologies. In contrast, the proposed interface can provide intuitive understanding by taking advantage of user interactions. Especially, it can show the connectivity between fibers, their topological shapes, and their geometrical configuration in the domain. Such visualization can respect the conventional methods among domain experts, while offering new tools for providing further insight. For evaluating the interface, a user study was conducted, in which experts and learners of fiber topology verified the usability of the proposed exploration interface.

Finally, the proposed approach is applied to time-varying scalar fields, in order to extract the time-evolution of local extrema. Existing techniques cannot respect the continuity of local extrema in time, when deciding the importance of them for selection. In contrast, the proposed scheme overcomes this problem by setting the time as a variable to track the temporal behaviors of the fiber, and simplifies the obtained multivariate topology. An evaluation is given by applying it to fluid simulation datasets.

The entire work is evaluated by discussing the effect of quantizing the data values and the effectiveness of the approach. On one hand, the effect of

quantization is investigated by comparison between the visualization results with different resolutions. On the other hand, the effectiveness is given a discussion on generalizing the dimensions and on the degeneracies of topological changes.

# Acknowledgements

I would like to express my sincere gratitude to kind support on this work from many people, although a complete list is impossible to insert herein.

First of all, I am grateful to Professor Shigeo Takahashi, who supervised me for five years with great patience, while providing me with a great opportunity to work in the research field of visualization. I thank Professor Osamu Saeki for his suggestions and directions among other things, which were substantial. My appreciation extends to Professor Hamish Carr for his various supports, including precious advice on adapting to visualization research.

I appreciate the extensive support of the members in the former Takahashi research group. I am especially thankful to Hsiang-Yun Wu for her kind help. Chongke Bi and Keisuke Kikuchi spared no effort to collaborate. I also thank Professors Takahiro Yamamoto and David Duke for genuine assistance. Students and collaborators of Professor Saeki kindly participated the user study. Especially, Masaya Sato generously volunteered for the preparation. Masayuki Kawashima's comments greatly enhanced the quality of the present work and inspired me with future directions. I appreciate Professors Akira Ejiri, Issei Fujishiro, Takehiko Sasaki, and Masashi Sugiyama for warm and indispensable comments as my dissertation committee.

Further, I thank Amit Chattopadhyay, Zhao Geng, and the members of Sugiyama research group, for their various forms of help. Japan Atomic Energy Agency supported me by recruiting me as a special research student. Especially, Hiroko Miyamura, Yasuhiro Idomura, Shinya Maeyama, Kimiaki Saito, Takuma Kawamura, Akiyuki Seki, Chika Kawai, and Hiroshi Takemiya kindly volunteered for invaluable helps. In specific, Shinya offered datasets for analysis. I thank Kenji Ono for offering datasets, and RIKEN for the permission. I also thank Peer-Teemo Bremer and Julien Tierny for providing datasets.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Fiber Topology in Visualization . . . . .	2
1.1.1	Fiber Topology for Scalar Functions . . . . .	5
1.1.2	Fiber Topology for Multivariate Functions . . . . .	5
1.2	Motivations and Challenges . . . . .	6
1.3	Objectives . . . . .	7
1.4	Contributions . . . . .	8
1.5	Organization . . . . .	9
<b>2</b>	<b>Related Work</b>	<b>11</b>
2.1	Fiber Topology in Scalar Fields . . . . .	11
2.1.1	Fiber Singularities . . . . .	11
2.1.2	Fiber Tracking: Reeb Graphs and Contour Trees . . . . .	12
2.2	Fiber Topology in Multivariate Data . . . . .	13
2.3	Fiber Topology in Time-Varying Scalar Fields . . . . .	15
<b>3</b>	<b>Extracting Singular Fibers</b>	<b>18</b>
3.1	Singular Fibers . . . . .	18
3.2	Detecting Fiber Topology . . . . .	25
3.2.1	Representing Fibers with the Joint Contour Net . . . . .	25
3.2.2	Identifying Singular Fibers . . . . .	27
<b>4</b>	<b>A User Interface for Studying Singular Fibers</b>	<b>33</b>
4.1	Tasks for Studying Singular Fibers . . . . .	34
4.1.1	Drawing the Fibers . . . . .	40
4.1.2	Perturbing the Function . . . . .	41
4.2	The User Interface . . . . .	41
4.2.1	The Domain View . . . . .	42

4.2.2	The Range View . . . . .	43
4.2.3	The Reeb Space View . . . . .	45
4.2.4	Function Perturbation . . . . .	45
4.3	Techniques . . . . .	46
4.3.1	Placing Inset Views . . . . .	46
4.3.2	Extracting Isosurfaces and Fibers . . . . .	47
4.3.3	Perturbing the Function . . . . .	47
4.3.4	3D Layout of the Reeb space . . . . .	48
4.4	Implementation . . . . .	51
4.5	Outcomes . . . . .	52
4.5.1	Task Support . . . . .	52
4.5.2	Informal User Study . . . . .	62
<b>5</b>	<b>Simplifying the Time-Evolution of Local Extrema</b>	<b>65</b>
5.1	Analyzing the Time-Evolution of Local Extrema . . . . .	65
5.2	Objectives . . . . .	66
5.3	Visualization Techniques for Tracking the Time-Evolution of Local Extrema . . . . .	69
5.4	Mathematical Formulation . . . . .	71
5.4.1	Time-varying Scalar Fields and Their Jacobi Set . . . . .	71
5.4.2	Simplification through Modifying Fiber Topology . . . . .	73
5.4.3	Persistency . . . . .	75
5.5	Simplifying Local Extrema . . . . .	75
5.5.1	Obtaining the Connectivity of Fibers . . . . .	76
5.5.2	Constructing the Joint Contour Net . . . . .	77
5.5.3	Simplifying the Jacobi Set through Reeb Skeleton Sim- plification . . . . .	77
5.5.4	Extracting the Jacobi Set . . . . .	78
5.6	Results and Evaluation . . . . .	79
5.6.1	A Simple Fluid Dataset . . . . .	79
5.6.2	A Fluid Dataset with Abundant Vortices . . . . .	80
5.6.3	An Analytic Function . . . . .	80
<b>6</b>	<b>Discussion</b>	<b>107</b>
6.1	Resolution of the Quantization . . . . .	107

6.1.1	The Resolution and Identification of the Topological Shapes of Singular Fibers . . . . .	108
6.1.2	The Resolution of Time . . . . .	108
6.1.3	The Resolution of Scalar Field . . . . .	112
6.2	Classification for Errors in Extracted Singular Fibers . . . . .	112
6.3	Miscellaneous . . . . .	128
<b>7</b>	<b>Conclusion and Future Work</b>	<b>131</b>



# Chapter 1

## Introduction

Nowadays, the spatiotemporal distribution of variables is analyzed in various scientific and engineering tasks. Often, such distribution not only has high complexity and ever-increasing resolution [MCC<sup>+</sup>99, GDS<sup>+</sup>06], but also has a large number of variables [WB97, DOL03]. For instance, a climate dataset can contain as variables many physical quantities, including temperature, pressure, and humidity. Such data, which has multiple variables, is called *multivariate data*, and is widely analyzed in a large number of research areas. In order to assist the exploration of a single variable, the topology of inverse images, in short *fiber topology*, is widely used. Fiber topology can provide both local and global information of a distribution. (See Figure 1.1.)

Although the analysis of fiber topology has primarily been applied to scalar data, fiber topology is inherent also in multivariate data. Given the importance of finding correlations between variables, extending the techniques for scalar data to multivariate data is pursued, in order to extract local and global information of distribution. This chapter starts by introducing how fiber topology has been applied in the context of visualization. Then, explanation is given for the motivation and objectives to extract, explore, and apply fiber topology for visualizing multivariate data.

## 1.1 Fiber Topology in Visualization

In order to understand the distribution of data values, visualization techniques for directly showing the distribution, such as isocontouring [LC87, Nie03] and volume rendering [DCH88] are widely used. Here, datasets easily have such a complexity that finding useful information is difficult without selecting informative parts from the entire data. One common tool for such a selection is fiber topology, which has offered efficient computational tools.

Fiber topology regards spatiotemporal distribution of values, which are either scalar or multivariate, as a function  $f : D(M) \rightarrow R(M)$ . Here,  $D(M)$  is the *domain manifold* such as the 3D space, in which the sampling points are distributed.  $R(M)$  is the *range manifold* such as a 1D space, representing the data values. In this thesis, both  $D(M)$  and  $R(M)$  are Euclidean spaces of some dimensions, since this is the situation for most data analysis. An inverse image of  $f$  is called a *fiber*, and fiber topology refers to the connectivity of the points in the inverse image  $f^{-1}(r)$  of some value  $r \in R(M)$ , and the connectivity of the inverse images across different function values. As depicted in Figure 1.1, fiber topology provides a way to encode and decode singularity and topology of inverse images.

We further introduce the fiber topology-based visualization in the following manner. We begin with the scalar topology, which has matured in visualization and has various applications. Secondly, we introduce the recently developed multivariate topology, which is the main topic of this thesis. Here, the topological analysis are generalized to multivariate functions. Roughly speaking, the research on multivariate cases follows the history of its scalar counter part, as depicted in Figure 2.1, and described in the followings.

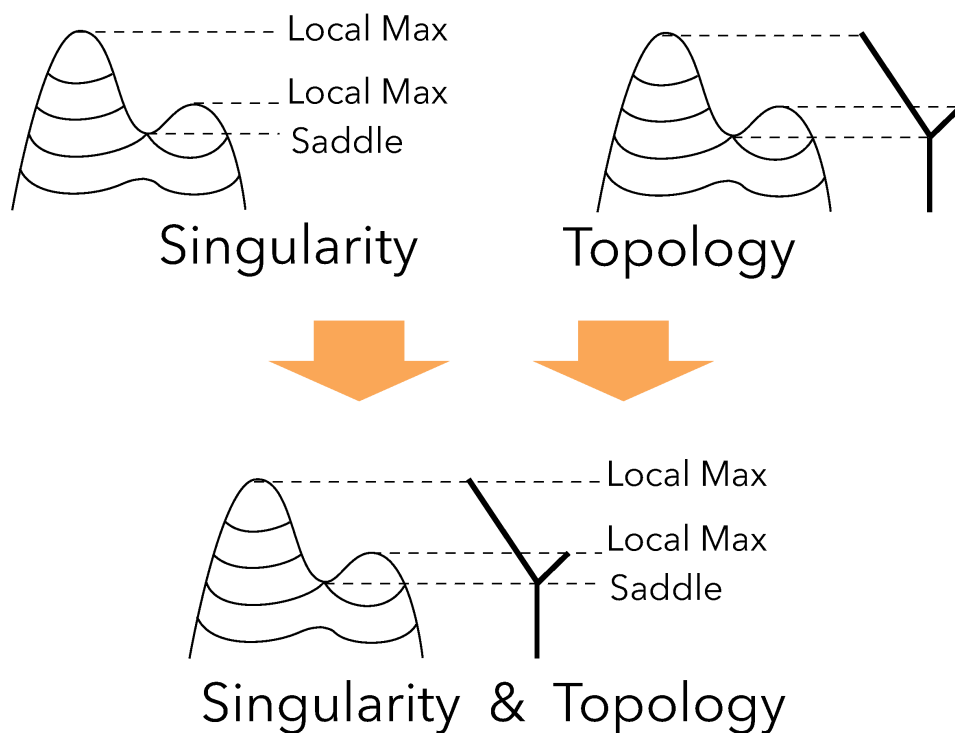


Figure 1.1: Fiber topology in univariate data provides the singularity and the topology of inverse images. Both the singularity and the topology benefits us for understanding data. As an example, we take the exploration of a terrain data. While the singularity describes the local features such as locations corresponding to local extrema or saddles, the global informations such as the configuration of the mountains is not apparent. Topology complements the exploration by encoding that two peaks are connected at a saddle.

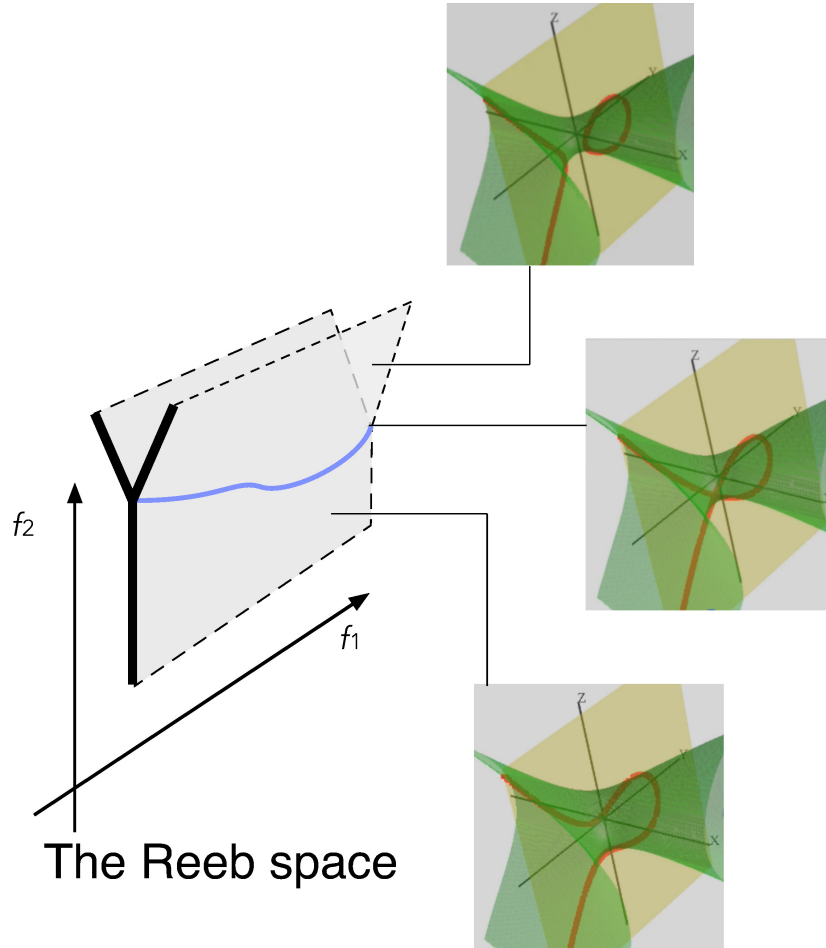


Figure 1.2: An example for the Reeb space of a function  $f(x, y, z) = (f_1, f_2)$ , as an extension of Figure 1.1. A point in the Reeb space corresponds to a connected component of a fiber. The connectivities of the points encode the connectivities of the fibers. Here, fibers are drawn as the red curves in the labelled figures. The yellow surfaces are the isosurfaces of a data value  $f_1$ , rendered in the 3D domain. On the other hand, the green surfaces are the isosurfaces of  $f_2$ , and the fibers can be identified as the intersections of the isosurfaces of  $f_1$  and  $f_2$ . With a low  $f_2$  value, only a single connected component exists, for each function value. At points in the blue curve, the component is splitting to two. In the upper part of the Reeb space, two points exist for a pair of value  $(f_1, f_2)$ . This corresponds to the fact that two connected components exist for such a pair.

### 1.1.1 Fiber Topology for Scalar Functions

For the analysis of a scalar field, one often tracks the isosurface evolution according to the scalar field value. Exploring such an inverse image of the scalar function allows us to effectively encode the singularities in the scalar field. This has often been accomplished by characterizing the topological transitions of such isosurfaces with the graph representations such as Reeb graphs [Ree46]. Indeed, studies on the topology of isosurface attracted not only learners in mathematics, but also researchers in the field of scientific visualization since we can enjoy a lot of benefits from it, for systematic exploration of isosurfaces [BPS97, CSA03] and transfer function design [TTF04, WDC<sup>+</sup>07], for example. Thus, tracking the inverse image of the scalar field function has been one of the promising approaches to investigating important features embedded in the given 3D scalar fields.

### 1.1.2 Fiber Topology for Multivariate Functions

However, extending scalar topology to the multivariate case is not straightforward. It is challenging firstly in the sense of the underlying mathematics, and secondly of the development of computational tools.

From the mathematical perspective, the difficulty is that the interactions of fibers are more complex than those in scalar topology. Indeed, in mathematical studies for fiber topology, the smallest interesting case maps 3-manifolds to 2-manifolds, and more recent work [Sae04] maps 4-manifolds to 3-manifolds. Given that visualization researchers are interested in arbitrary dimensionality of the target manifold, i.e. an arbitrary number of variables to be analyzed, necessary mathematics inevitably involves spaces of higher dimension, which are beyond the capability of visual perception.

On the other hand, from the perspective of developing computational tools, the study has begun only recently. Although the Reeb graph [Ree46]

was extended to Reeb space [EHP08] in 2008, its practical computation [CD14] has become available only in 2013. Still, direct application of this new computational tool is currently not fully developed. While in the scalar case the nodes and edges of a Reeb graph directly correspond to features such as the boundary of objects, suitable features that can be analyzed with the Reeb space are to be found. In this thesis, we set the singularity of fibers as the features. For further explanation, we begin with explaining our motivations and challenges.

## 1.2 Motivations and Challenges

As we have seen in the preceding section, fiber topology had the ability to efficiently encode the distribution of data values, for scalar data. Technically, the following challenges exist for analyzing fiber topology of multivariate data.

The most important challenge is the degree of freedom. In the scalar case, although data consisted of discrete samples, the connectivities of fibers between interpolated values could be fully computed. The computation was achieved by locating the sampling points in a 3D mesh, which defined the linear interpolation of data values. Then, the connectivity of the fibers was tracked along the edges of the 3D mesh. Here, this one-dimensional tracking could extract the connectivities between fibers of neighboring scalar values, since the neighborhoods were also one-dimensional manifolds.

In the multivariate case, however, the tuple of variables varies in multiple degrees of freedom. Hence, as depicted in Figure 1.2, the connectivity of fibers forms a space of multiple dimensions, which is the *Reeb space*. Therefore, tracking the connectivity among fibers should also be done in multiple degrees of freedom. Here, the conventional approach in the scalar topology analysis fails. For an explanation, recall that the approach in scalar topology

was to track the sampling points, along the edges of a 3D mesh. However, such a traversal is a one-dimensional motion in the multi-dimensional neighborhood of data values. Therefore the multi-dimensional connectivity of fibers cannot be fully obtained. To overcome this problem, Carr and Duke [CD14] quantized the data values. Here, thanks to the quantization, we can track the connectivity between fibers having neighboring data values. This is achieved by tracking the data values in the range along the directions of each axis of the data values. However, the challenge remains as to how we should extract the information of fibers other than their connectivity, and how we should treat the quantized data values, instead of the row values that were conventional in visualization techniques based on fiber topology.

Additionally, we find the following challenges. Firstly, as opposed to the continuous analytic functions, the discrete model of fiber topology requires a strategy to handle the behavior in numerical data, such as errors in data values. While mathematical study defined a classification with ideal assumptions, it is rarely valid in numerical situations. Secondly, suitable applications should be found. Although the scalar topology has found various applications, the extension of their problems has not been found. Indeed, this is a mathematically involved problem and researchers in visualization and mathematics are collaborating to resolve this situation [STS<sup>+</sup>14]. Handling realistic datasets differs from analytic functions, in the sense that overwhelming amount of topological features exists, often due to the complex physical setup, or/and due to the noise in measurements. Therefore, special care must be taken for reducing the features shown to the user.

### 1.3 Objectives

The objectives of this thesis are to extract, and apply fiber topology for multivariate functions  $f : \mathbf{R}^n \rightarrow \mathbf{R}^m$ . We choose multivariate functions

$n = 3$  and  $m = 2$ , since it is the simplest yet interesting case considering fiber topology. The central idea is to analyze the connectivity of the fibers via the Reeb space (Figure 1.2), which is an extension of the Reeb graph. While Carr and Duke [CD14] approximated the Reeb space with the joint contour net, extracting the singularity of the fibers has remained as a challenge. We extract the singularity of fibers as the singular fibers [Sae04], and apply them to visualization. In the next section, we explain the contributions of the work.

## 1.4 Contributions

The present work has the following major contributions:

- Extracting singular fibers
- A user interface for studying singular fibers
- Simplifying the time-evolution of local extrema

We extract the topology of fibers [SSC<sup>+</sup>] and apply them to exploring multivariate data [SSC<sup>+</sup>] and real datasets.

Firstly, in Chapter 3 we characterize topological features as singular fibers [Sae04]. We not only identify the original classification in mathematics [Sae04], but also additional topological features introduced by the boundaries of the domain. Here, it is worth mentioning that we asked our collaborators to extend the classification, which led to a new result in mathematics [SY14]. Furthermore, we newly developed a hierarchical identification algorithm of singular fibers that firstly identifies all of them, and then further categorizes theoretically ideal ones. Fiber topology is extracted by analyzing the Reeb space [EHP08] through its approximation by the joint contour net [CD14]. Secondly, in Chapter 4, we apply the approach to an interface for mathematical studies. With our interface, the user can analyze fiber topology in functions that are too complex to analyze manually. This interface respects



and extends the conventional manual visualization by the domain experts. Finally, we apply fiber topology to analyzing realistic data, which are the local extrema in time-varying scalar fields. In contrast to the existing methods that make use of topology, which had difficulty tracking the connectivity of fibers along the temporal axis, the presented approach tracks such connectivity by making use of our singular fiber extraction scheme in Chapter 3.

## 1.5 Organization

As shown in Figure 1.3, the remainder of the thesis is organized as follows. In Chapter 2, the related work is presented. Then, we present the contributions of this work, which are divided into 3 chapters. Chapter 3 presents the proposed extraction and identification scheme for singular fibers. Chapter 4 introduces the proposed interface for studying fiber topology in mathematics. Chapter 5 presents our idea of applying fiber topology to real world datasets, which are the visualization of local extrema in time-varying scalar fields. We evaluate the work in Chapter 6. The conclusion and future work are presented in Chapter 7.

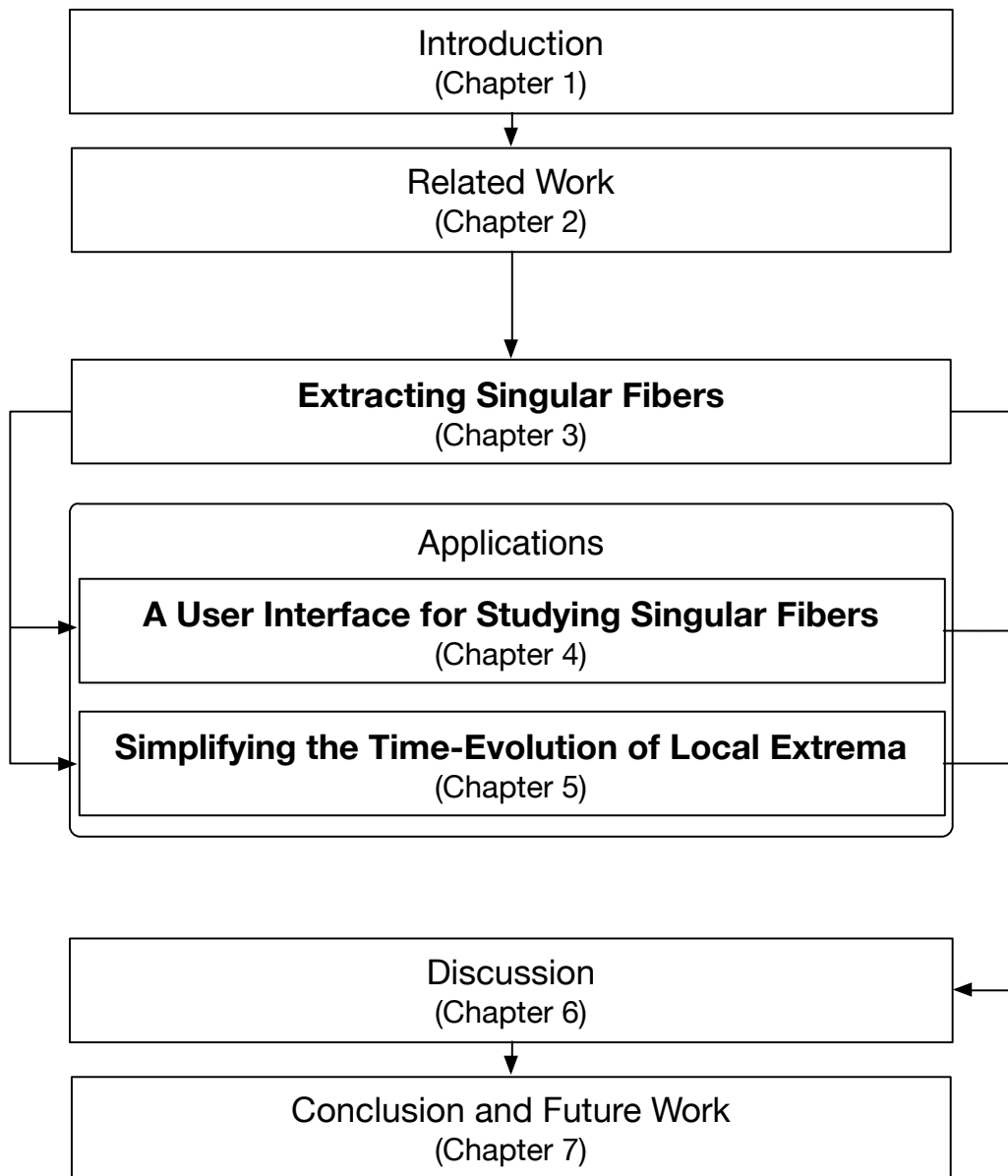


Figure 1.3: The organization of this thesis.

# Chapter 2

## Related Work

This chapter presents the related work. Firstly, we introduce how fiber topology has been applied to visualizing scientific data, in terms of scalar fields and multivariate fields. Then, we explain the conventional methods which relate to the proposed application of fiber topology to visualizing time-evolution of critical points.

### 2.1 Fiber Topology in Scalar Fields

Analysis of scalar field is an active research area in scientific visualization. There exist various computational tools for understanding the spatiotemporal distribution of data values, including isocontouring [Nie03] and volume rendering [DCH88]. Discrete data is often complex since it is full of features due to noise and complexity of the phenomena. To deal with the enormous amount of singularities, a promising approach is topological analysis [Ede09, PPF<sup>+</sup>09].

#### 2.1.1 Fiber Singularities

Weber et al. [WSHH02, WSH03] proposed approaches to emphasize the isosurfaces of critical values to efficiently explore useful features in scalar fields. Note that isosurfaces can be regarded as singular fibers in scalar fields [Sae04].

### 2.1.2 Fiber Tracking: Reeb Graphs and Contour Trees

Bajaj et al. [BPS97] proposed contour tree, which represents the topology of contours as a graph with nodes encoding critical points such as local extrema and saddles, and edges encoding the connectivity of contours. Contour trees are not only useful for describing differential features such as local extrema and saddles, but also for extracting structural information like inclusion of a small hill in a mountain. While contour trees are defined for simple domains, i.e. a finite space without holes, they can be extended also for non-simple domains. Their general counter part is Reeb graphs [Ree46], which are not necessarily trees but still graphs. Reeb graph was originally introduced to describe the structure of terrains [Ree46]. Applications of Reeb graph as a computational tool originate in the description of the terrain [BR63, FM67]. A variety of applications then appeared in image processing [SC86] and GIS [TIS<sup>+</sup>95]. In scientific visualization, examples include designing transfer functions of volume rendering [FTAT00, WSHH02] and 3D shape model retrieval [HSKK01]. Van Kreveld et al. [vKvOB<sup>+</sup>97] proposed an algorithm for 2D and 3D scalar fields. The complexity is  $O(M \log M)$  and  $O(M^2)$ , where  $M$  is the number of elements in a simplicial mesh, for 2D and 3D, respectively. Tarasov and Vyalyi [TV98] reduced the complexity to  $O(M \log M)$  for the 3D case. Eventually, Carr et al. [CSA03] simplified the algorithm and made it work for arbitrary dimension. As mentioned in Section 1.2, the computation of contour trees is based on tracking contours along the scalar values. This is done by sorting the sampling points according to the scalar values, and then iterating through the sampling points. Note that this is not directly extensible to multivariate topology, where one has to track the fiber in multiple directions of multiple variables, instead of a unique direction in the conventional case of scalar fields. Later, Pascucci et al. [PSBM07] proposed an algorithm for constructing Reeb graphs, which are

the generalization of contour trees.

Contour tree was then enhanced with additional informations such as iso-surface genera [PCM02] and spatial embeddings [TTF04, TFT05, TTFN05], topology on boundaries [CS09] and employed as visual metaphors of the given datasets [WBP07]. Since then, contour trees were used for semantic segmentation of the entire 3D scalar field [CS03], then extended to portray topological complexity in a time-varying data set by counting connected components as isovalue and time varied, then presenting the information as a color map for visual analysis [KRS03]. These approaches have also been used to reduce very high dimensional data to two or three for visualization [TFO09, OHJ<sup>+</sup>11, GBPW10].

## 2.2 Fiber Topology in Multivariate Data

Usually, multivariate functions in the visualization community are multivariate datasets, for which each field value is sampled at spatiotemporal coordinates. Multivariate functions are usually more difficult to analyze than scalar fields, so fewer effective techniques are known [FH09]. Various approaches have been attempted, including combining several techniques such as iso-contouring and arrows [Tre00], multi-dimensional transfer function design [KKH01], and information theory [BDSW13].

Subsequently, topology came into the roll. Bachthaler and Weiskopf [BW08] extended conventional scatterplots to the continuous version, projecting smooth density within the domain onto the range. Lehmann and Theisel [LT10] extended this work by identifying discontinuities as important features, and it is clear at present that these features are identical to the boundaries identified by singular fiber topology.

An alternate approach to multifields analyzes contour trees of individual fields, then compares the features of each [SWC<sup>+</sup>08], but this is not the

same as those analyzing the Reeb space. Equally, Edelsbrunner et al. [EH04] extracted the Jacobi set from discrete samples of two scalar field functions, but did not extract the entire Reeb space. Instead, the Reeb space was formulated later [EHP08], and can now be seen to be identical to the Stein factorization [Lev85]: in neither case was a practical computation given.

The Jacobi set has since been simplified [BWN<sup>+</sup>13, SN11] and applied to ridge-valley extraction [NB13]. However, in these cases, the Jacobi set is illustrated in the  $\mathbf{R}^2 \rightarrow \mathbf{R}^2$  case, where the Jacobi set divides the domain into regions. For higher dimensions, the region is actually separated into regions by the singular fibers which pass through points of the Jacobi Set. In the  $\mathbf{R}^2 \rightarrow \mathbf{R}$  case, the Jacobi set is the set of critical points (extrema and saddles), but the critical contours must be extracted to identify features of uniform topological behavior. Correspondingly, in the  $\mathbf{R}^3 \rightarrow \mathbf{R}^2$  case we are considering, the Jacobi set is a set of 1-manifold structures, as can be seen in Bremer et al. [BBD<sup>+</sup>07, Fig. 5(a)]. It then follows that not only the Jacobi set needs to be computed to support the mathematical workflow, but also the Reeb space.

More recently, Carr and Duke approximated the Reeb space with the *Joint Contour Net* (JCN) [CD14]. Here, each dimension of the range space is partitioned into intervals of fixed height and width (i.e. pixels) and projected onto the graph of the function. Each pixel corresponds therefore to a quantized approximation of a fiber, and this permits extraction of the fiber components and their relationships.

The advantage of the JCN is that it explicitly encodes the connectivity of fiber components as a graph structure, albeit approximately. We therefore use it to extract the fiber topology of the function  $\mathbf{R}^3 \rightarrow \mathbf{R}^2$  within the original finite 3D domain and separately for the restriction to the boundary.

Visually, Duke et al. [DCK<sup>+</sup>12] arranged JCNs aesthetically in 3D space for arbitrary dimension of the range space, and applied it to the analysis of

real datasets. In this application, however, it is better to keep the connection with the range, and this requires a new layout algorithm.

## 2.3 Fiber Topology in Time-Varying Scalar Fields

In the scientific visualization community, isocontouring has been applied to investigate the distribution of data values in time-varying scalar fields [MS09]. Another approach for time-varying scalar fields in scientific visualization is activity detection. Computer vision [Bal82] has independently developed methods to track time-varying features in 2D domains, and they were applied for tracking features [SSZC94, ADM92, VV92]. However, the features of interest have distinct behaviors such as appearing, splitting, merging, or disappearing, which do not happen in computer vision. For the sake of such specific feature extraction in scientific visualization, Silver and Wang [SW97, SW98] tracked features that are defined as regions in the domain that satisfied predefined criteria. The features in different time steps were checked for their overlaps to determine their continuity over time. However, this resulted in the disconnections of features that move fast, without an overlap.

Recently, topological analysis has been used for tracking objects. Bremer et al. [BBD<sup>+</sup>07] visualized the critical points in time-varying scalar fields, and simplified the results. In their approach, contours that lasted long can be misleadingly removed if their spatial persistency was small. This happened as the algorithm did not extract the connectivity of the critical points. In contrast, approaches based on Reeb graphs [JSW03, SB06] could obtain the life span of the critical points, but the topology of the fibers was still tracked only for spatial axes and not time. In comparison, our approach interpolates the scalar values in all of the spatiotemporal directions. In addition, in the

Reeb graph based approach mentioned above, the user had to select the isovalue to be visualized. Therefore, the life spans of contours were not pre-computed. Our approach does not have this problem, since the JCN explicitly maintains the correspondence between all critical points across different time steps.

Furthermore, visualization techniques are known for visualizing flow fields, although they are not directly related to fiber topology. Therefore, reviewing them in the context of fiber topology is misleading, and we choose to give a concise review in Chapter 5 together with their own context and necessary background on time-varying 2D scalar field visualization. In the succeeding Chapter 3, we will introduce the supporting technique: the proposed approach for extracting and identifying the singular fibers from multi-fields.



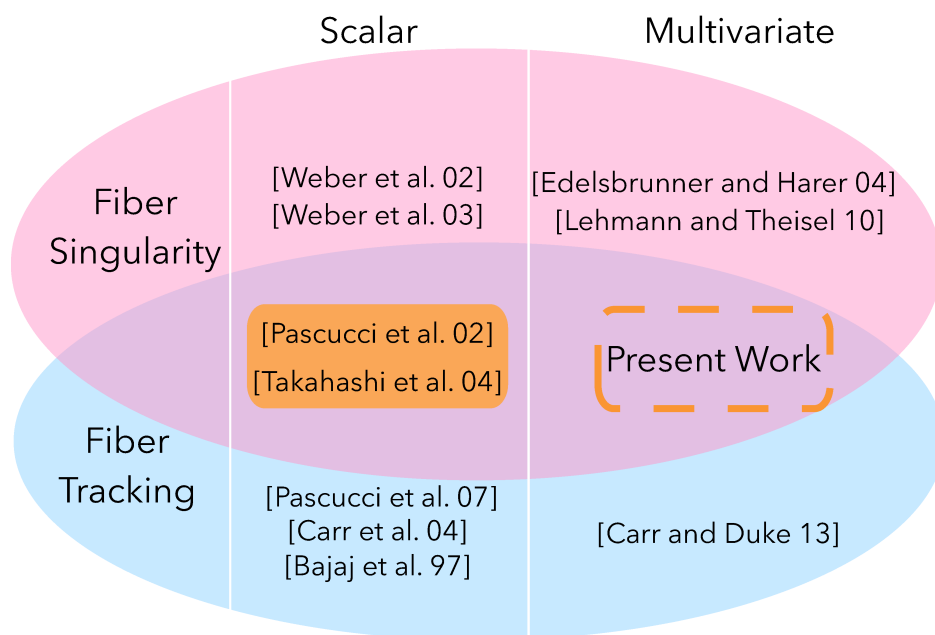


Figure 2.1: Work on extracting fiber topology and singularity at the same time together with their correspondence.

# Chapter 3

## Extracting Singular Fibers

This chapter presents the definition of the types and the shapes of fibers, as well as the detection scheme of the topological shapes of the fibers. We start by introducing mathematical backgrounds that are necessary for understanding fiber topology, and the limitations in visualizing numerical datasets. After them, the visualization-specific classification and detection follow.

### 3.1 Singular Fibers

Singular fiber theory [Sae04] studies mappings from manifolds of arbitrary dimension to other manifolds of dimension less than the domain, often exactly one dimension less than the domain. We consider a common case in multi-field data analysis: simple manifolds, with small dimensions of functions  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ , as this is challenging, but can still be displayed in a 3D interface, and develops intuition for higher-dimensional cases. Refer to [STS<sup>+</sup>14] for a formal introduction to the mathematical backgrounds.

We assume that a function  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^2$  where  $f$  consists of 2 scalar functions  $f_1$  and  $f_2$ . The *function value*  $f(\mathbf{x})$  refers to the vector  $(f_1(\mathbf{x}), f_2(\mathbf{x}))$ . For any point  $\mathbf{c}$  in  $\mathbf{R}^2$ , the *fiber* of  $f$  at  $\mathbf{c}$  is defined as the inverse image  $f^{-1}(\mathbf{c})$ . A *fiber component* refers to a connected component of a fiber.

As an example, consider Figure 3.1. Here, the planes are isosurfaces of

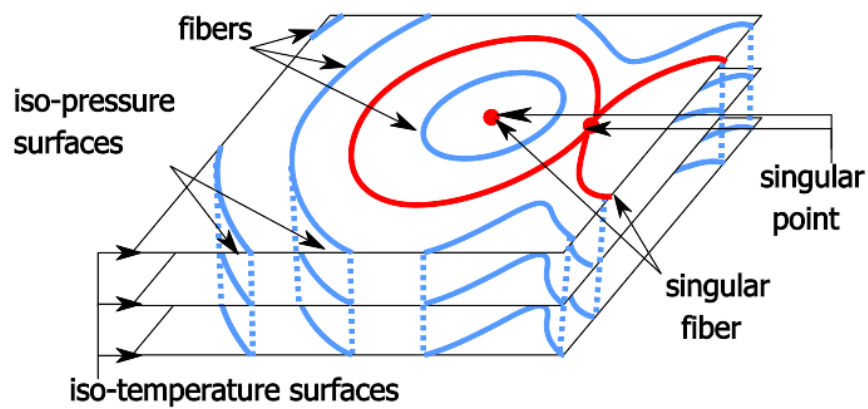


Figure 3.1: A singular fiber of a bivariate function. Isosurfaces of temperature are horizontal planes, while isosurfaces of pressure are curved sheets perpendicular to the planes. Fibers (i.e. inverse images) of the function are curves obtained by intersecting isosurfaces of temperature and pressure.

temperature, while the blue stacked curves belong to isosurfaces of pressure. For a particular pressure and temperature, we find the fiber by intersecting the pressure isosurface and temperature isosurface. Since these isosurfaces are 2-manifolds, most of the fibers are 1-manifolds embedded in  $\mathbf{R}^3$ , and we will see similarities to familiar contour lines. Indeed, readers familiar with 2D time-varying scalar field may regard it as a special case where  $f_2$  is the time axis.

Exploring the example in Figure 3.1, each non-empty fiber consists of one or more disjoint components. In this example, most fiber components are arcs or loops: arcs intersect the boundary twice, and are called *boundary fiber components*, while loops are in the interior of domain and are called *interior fiber components*.

We refer to this distinction as the *topological shape* of the fiber component, and to the set of topological shapes of fiber components as the topological shape of the fiber.

Fiber components whose topological shape is *not* an arc or loop are *singular*, and fibers with any singular component are also singular. The red fiber component in Figure 3.1 is such an example, and Figure 3.3 lists more examples. Any fiber consisting only of arcs or loops is then called *regular*.

As with isocontours, different range values give fibers with different topological shape, and singular fibers are found at values where topological shape changes. Unlike isocontour analysis, the value varies in 2D, so we cannot discuss changes as a single sweep through a 1D range. We instead take any path through the range: the topological shape must change smoothly along the path, and, for example, can change from a single loop to two loops at a *singular point* where the fiber takes on a figure-8 shape, as for example when crossing the blue curve in Figure 3.2. In this case, the fiber is of type V in Figure 3.3.

Returning briefly to scalar fields, we see that the critical values divide the

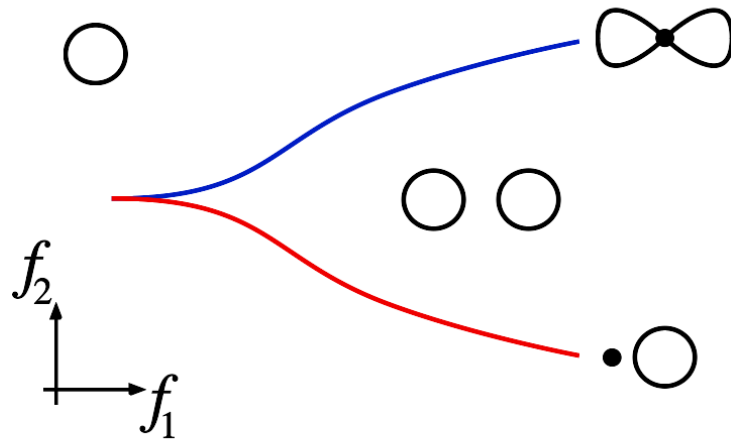


Figure 3.2: A typical hand-drawn fiber visualization: red and blue curves indicate singular values in the range. At the blue curve, a fiber component splits into two. At the red curve, a fiber component becomes a point and disappears. The topological shapes of fibers are represented by icons.

Type	Color	Topological change
0a	Grey	
0b	Grey	
I	Red	
II	Orange	
III	Yellow	
IV	Light Blue	
V	Green	
VI	Blue	
VII	Purple	

Figure 3.3: Topology of non-degenerate fiber components, after Saeki and Yamamoto [SY14]. For  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ , topological changes to fibers must be one of these possibilities or their converses, barring degeneracy. These are the same changes as for isolines in  $f : \mathbf{R}^2 \rightarrow \mathbf{R}^1$ , i.e. scalar critical points in a scalar field. However, since there are *two* range variables, these changes occur across linear boundaries in the range rather than in a strict linear sequence.

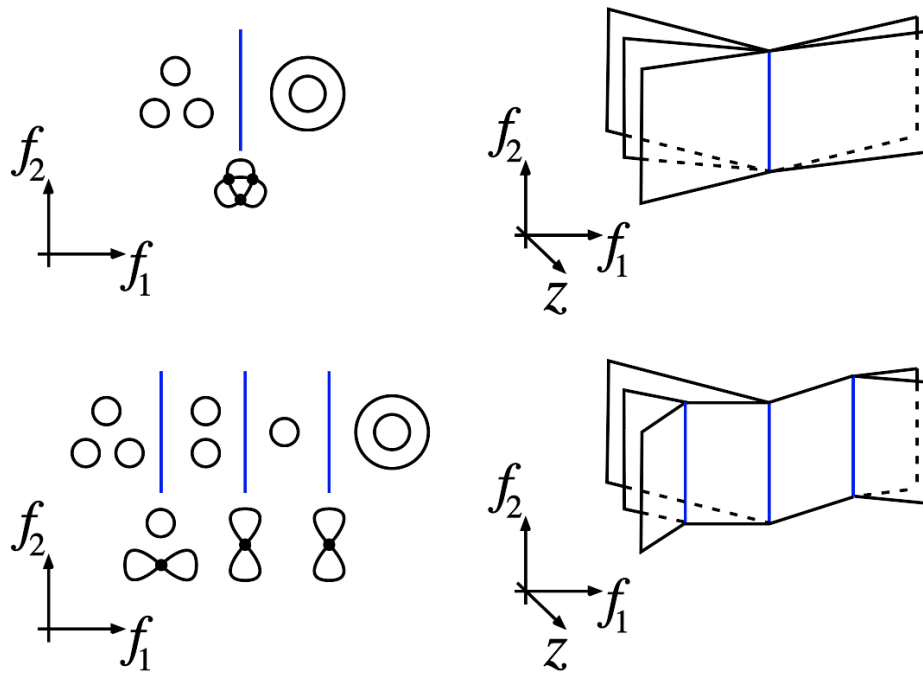


Figure 3.4: A degenerate fiber in a Reeb space. On the left, topological shapes of fibers shown in the range. On the right, the Reeb space, expanded along a virtual dimension  $z$  for the visual clarity. Above, three fiber components merge simultaneously at three singular points, along the singular values indicated by the blue line. Below, a perturbation to the function resolves the component to non-degenerate ones.

range interval into a set of intervals with different topology. We can track components separately, using overlapping intervals corresponding to edges of the Reeb graph. This has been formalized [CSA03] by defining edges to be the equivalence class of topologically similar contours that are adjacent to each other.

Applying the same logic here (shown in Figure 3.2), a region where all topological shapes consist of one loop must be divided from a region with two loops by a curve along which singular fibers of figure-8 shape occur (shown in blue) and by a curve along which one of the loops disappears (shown in red). This is again similar to the Reeb graph: at a critical point, one contour breaks into two, and the critical point is the boundary between having one contour and having two. At some higher isovalue, the two contours may either reconnect, or one may disappear entirely at a local maximum. In Figure 3.2, the red curve carries the semantics of a local extremum (i.e. extinction), while the blue curve carries the semantics of a saddle point (i.e. separation).

Singular fiber theory [Sae04] classifies singular fibers, and this classification gives the possible topological shapes in a fiber. In the past, singular fiber theory has assumed a function defined over the infinite domain without boundary, and it is only recently that it has been extended to deal with behavior at the boundary of a finite domain [SY14]. This is accomplished by classifying the topological shape not only of a fiber, but also of its restriction to the boundary of the finite domain, and was developed in part to support the application we describe, but the proof is beyond the scope of this thesis.

This classification is shown in Figure 3.3, and is the basis of our classification of singular fibers for visual display. Excluding degeneracies and cases involving boundaries, there are in fact two possibilities: birth and split, and the parallel with contours in the plane is apparent.

As with scalar analysis (Morse theory), the singular fiber theory assumes the functions to be stable, i.e. the topology of their fibers does not alter even



if a small perturbation is applied. Therefore, if the function is not stable, the topology of fibers can be different from those classified in [Sae04, SY14], and such fiber components are said to be *degenerate*. In particular, more than two fiber components can merge to one at a function value. An example is given in Figure 3.4.

The list in Figure 3.3 is not always easy to interpret in practice, as multiple topological changes can occur in a single fiber, especially for degenerate data. As a result, plotting a function of any complexity can lead to topological boundaries overlapping visually, requiring skilled interpretation of the interactions. We do not claim to resolve such situations, merely to accelerate the process of identifying them and manually resolving them through perturbation.

Note that the set of singular points is the *Jacobi set* of a function [EH04]. In addition, the *Reeb space* [EHP08], also known as the *Stein factorization* [Lev85] in mathematics, is the result of contracting each fiber component to a single point. Finally, note that the joint contour net [CD14] is a quantized approximation of the Reeb space, and will be discussed in more detail later.

## 3.2 Detecting Fiber Topology

As stated, we detect the singular fibers for stable functions.

### 3.2.1 Representing Fibers with the Joint Contour Net

Recall that the JCN is a tessellated approximation of the Reeb space. Each node in it corresponds to a class of fibers defined by the Euclidean product of intervals in the range. In the  $\mathbf{R}^3 \rightarrow \mathbf{R}^2$  case, these are rectangles in the range: i.e. pixels, and the JCN can be computed using rasterization algorithms [CD14]. A schematic picture of the process is given in Figure 3.5.

The resulting cubic lattice is then subdivided into six tetrahedra per cell

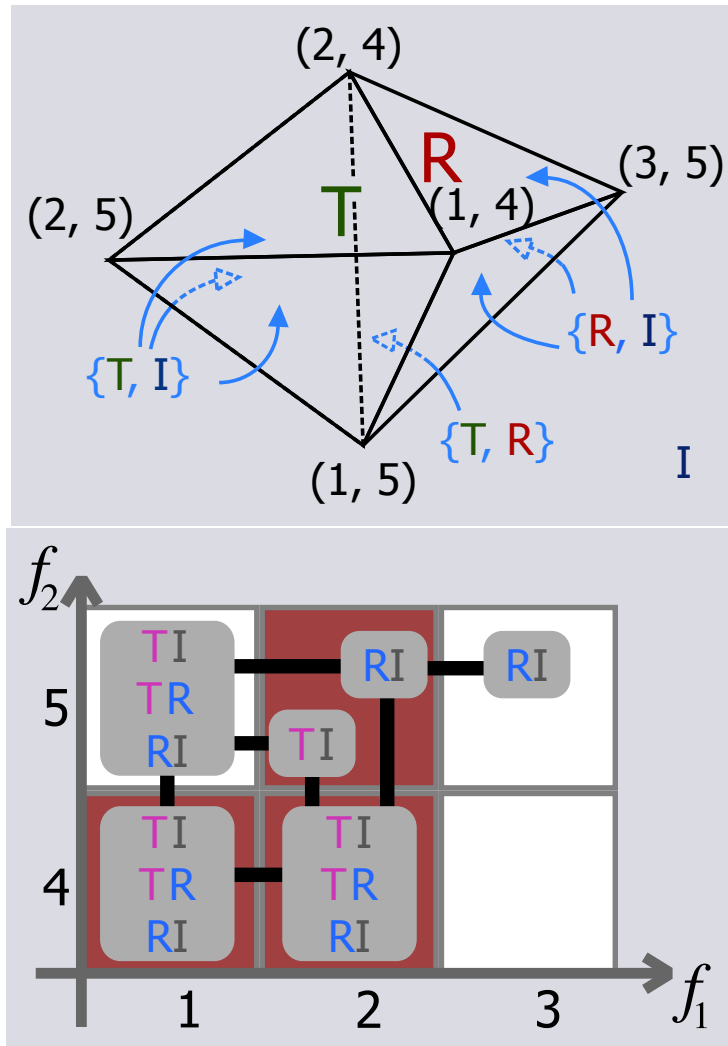


Figure 3.5: The construction of JCN for a function  $f: \mathbb{R}^3 \rightarrow \mathbb{R}^2$ .

and rasterized to compute the JCN, using colored fragments in the frame-buffer to represent slabs.

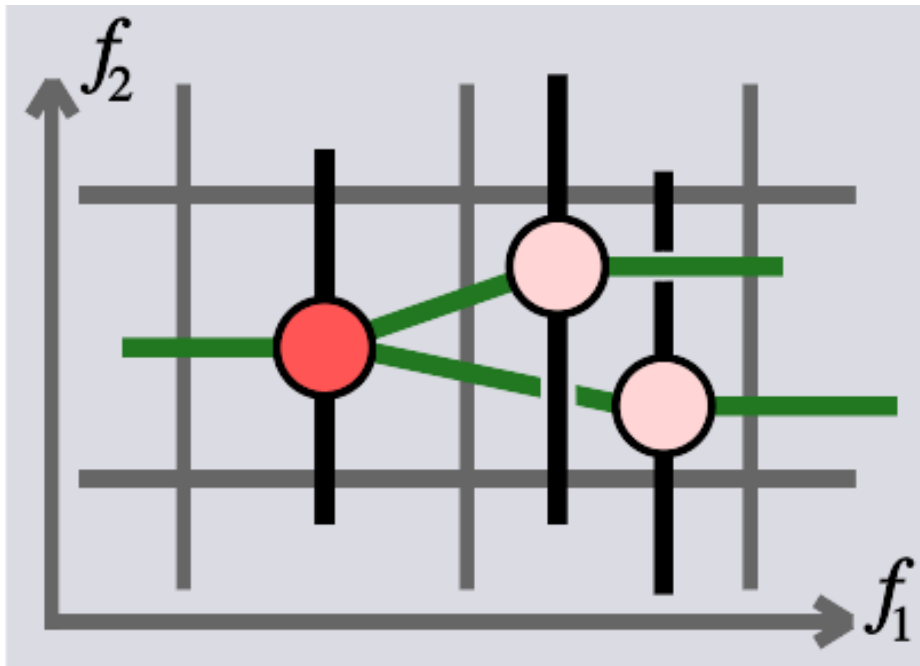
### 3.2.2 Identifying Singular Fibers

As seen in Section 3.1, the key task in singular fiber theory is to identify singular fibers and the topological changes that occur at them, as shown in Figure 3.3. We observe that each regular fiber component is represented by a single point in the Reeb space that is locally homeomorphic to a disk.

In the JCN, a node corresponding to regular fibers will be adjacent to exactly four other nodes: one each above, below, to the left and to the right, in the  $R^2$  range. Any node that does not have exactly these four neighbors is therefore presumed to be singular.

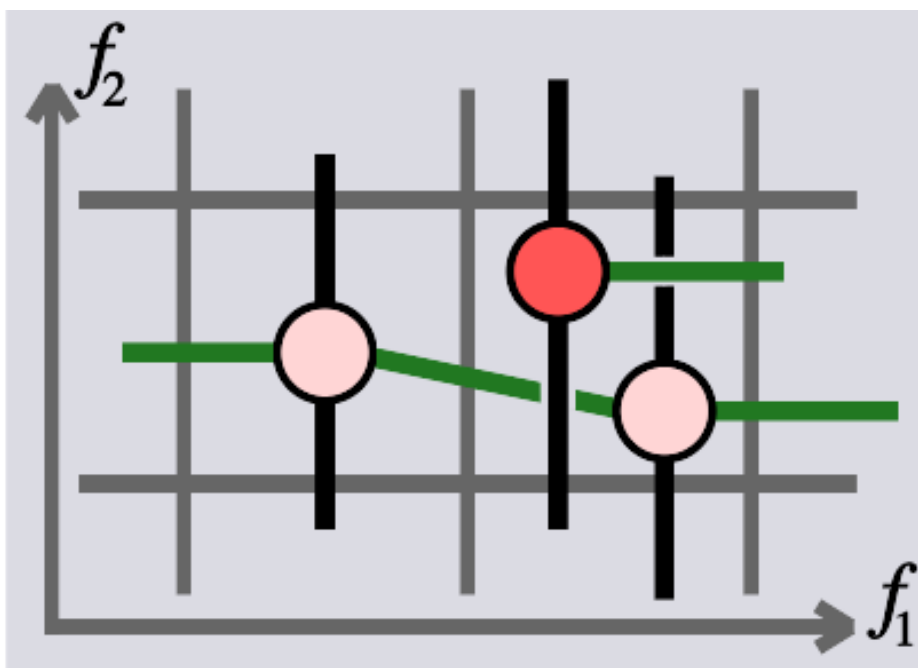
Even further, a singular fiber component's shape is deduced. Being inspired by Figure 3.3, we refer to the change in the number of components and the number of endpoints. Notice that we can determine whether a fiber component (i.e. node in the JCN) has a split (merge) or birth (death), as shown in Figure 3.6. More precisely, a fiber component where a split (merge) occurs will have at least two neighbors in at least one direction, while a fiber at which a new component is born (or dies) will have no neighbors in at least one direction. To discriminate the topological changes on the boundary of the domain, we construct not only the JCN for the entire domain (the interior JCN), but also the JCN of the function restricted to the boundary (the boundary JCN).

Then, the JCN is constructed. Following the rasterization algorithm of Carr and Duke [CD14], we construct the interior JCN by rasterizing all of the triangles for every tetrahedra in the tessellation. We accelerate the process by the hardware acceleration, while this was not discussed by Carr and Duke [CD14]. When constructing the JCN with a common graphics API such as OpenGL, a special attention must be taken to avoid degeneracy in



(a)

Figure 3.6: Identifying topological change (in red) from the JCN. (a) Split/merge of a fiber component.



(b)

Figure 3.6: Identifying topological change (in red) from the JCN. (b) Birth/death of a fiber component.

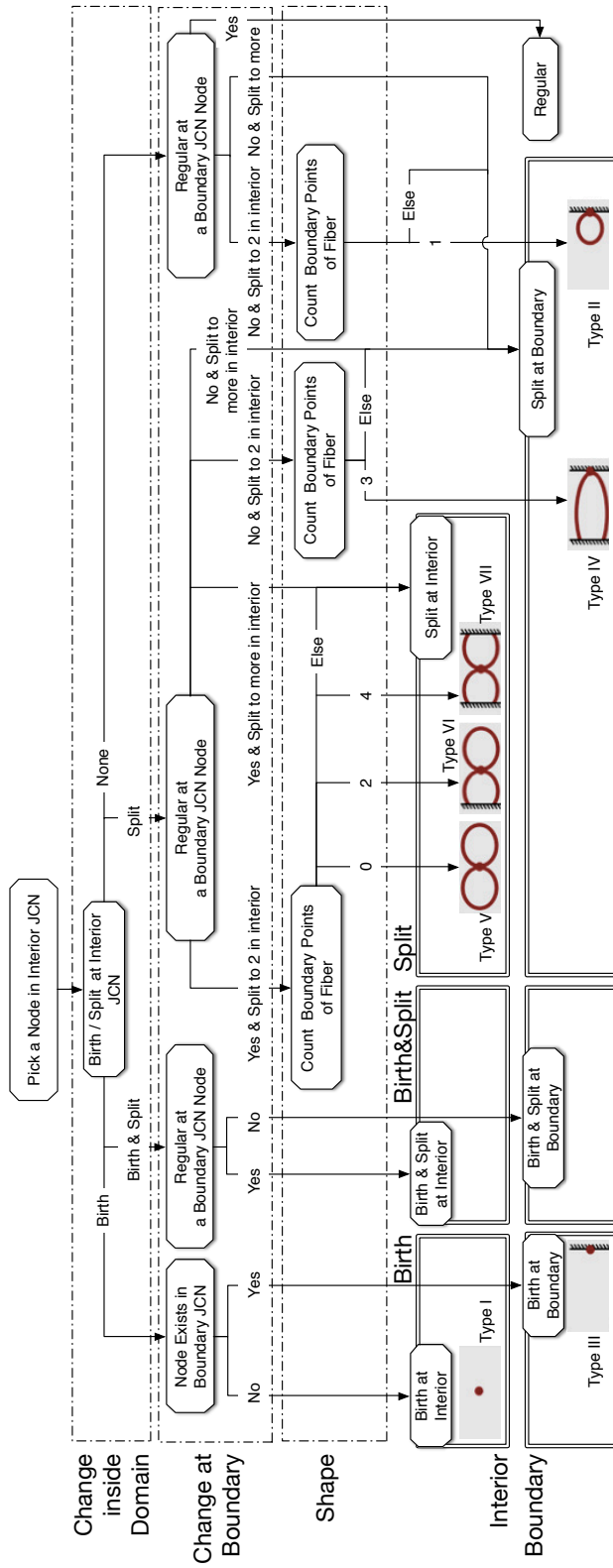


Figure 3.7: Decision tree for identifying singular fibers in the JCN, based on Figure 3.3. We identify not only the non-degenerate singular fiber components (shown by icons), but also the degenerate ones (with double lines). Birth/split detection is illustrated in Figure 3.6.

rasterization. In most graphics cards, if a triangle projects itself as a line or a point, the result is not drawn into the framebuffer and the JCN construction fails. To address this problem, we not only rasterize the triangle, but its boundaries, as line segments. Similarly, the line segment might be degenerate in the rasterization to find itself as a point. We therefore also rasterize the three points of every triangles. The triangle is then successfully rasterized as either line segments or a point, even in degenerate cases. Further, we group together two triangle fragments in the same pixel to form one JCN node, if the two triangles are both incident to a common tetrahedron as specified in the literature [CD14].

Note that a node can be regarded as a set of triangles. Likewise, the boundary JCN is constructed by rasterizing all the triangles on the boundary. Each node in the interior/boundary JCNs, then, corresponds to a slab (or more precisely, the set of tetrahedra/triangles that encloses a fiber component in the domain/boundary). For each fiber component that is encoded as an interior JCN node,  $i$ , its topology on boundary is obtained by referring to boundary JCN nodes, which are the set  $\{j \mid \text{a triangle in } j \text{ and another in } i \text{ share an incident tetrahedron}\}$ .

We iterate through the interior JCN nodes to determine the fiber component's topological shape: for each node, we consult the decision tree in Figure 3.7. The idea of the tree is to classify fiber components to three categories: singular ones with/without a critical point on boundary, and regular ones. The singular ones are further classified to "birth," "birth and split," and "split." In this way, any non-degenerate singular fiber component with a pattern in Figure 3.3 can be classified.

More complex non-degenerate singular fiber components have components being born or merged at multiple critical points. Unfortunately, discriminating these cases against cases in Figure 3.3 was not achieved, as this requires investigating the number of critical points. This requires a separate

work, since no critical point extraction algorithm is consistent with the approximation in the JCN, to the best of the author's knowledge. In practice, though, our identification results satisfied the users in Section 4.5.

The results of the approach is presented in Chapter 4, while detailed evaluation of the results is given in Chapter 6.



## Chapter 4

# A User Interface for Studying Singular Fibers

It is often assumed that mathematics supports visualization with concepts and equations to build upon. Yet there are also possibilities where computation can assist the development of mathematics. Many mathematicians construct visual representations by hand, which can be time-consuming and difficult, even for experts, especially in higher dimensions. Where the mathematical field is used to develop tools for analysis and visualization, there is therefore double value in building specialized tools for mathematical visualization.

One field where this is true is *fiber topology* – the multi-field counterpart of scalar Morse theory. Like in Morse theory, mathematicians study the relations between a function and the singularity of its fibers (i.e. inverse images), but for multi-fields, and formed the *singular fiber theory*. In the advancements of this theory, manual visualization has supported obtaining and communicating significant results. Therefore, semi-automatic visualization is advantageous both for experts and for learners to investigate phenomena that are too complex for manual visualization. As with the scalar case, topology of fibers in multi-fields is now being applied to data analysis and visualization [CD14]. Here again, manual visualization helps one communicate the mathe-

mathematical results to non-specialists, and develop the intuition of novices to the field. Therefore, there is considerable advantage in developing visualization systems for it.

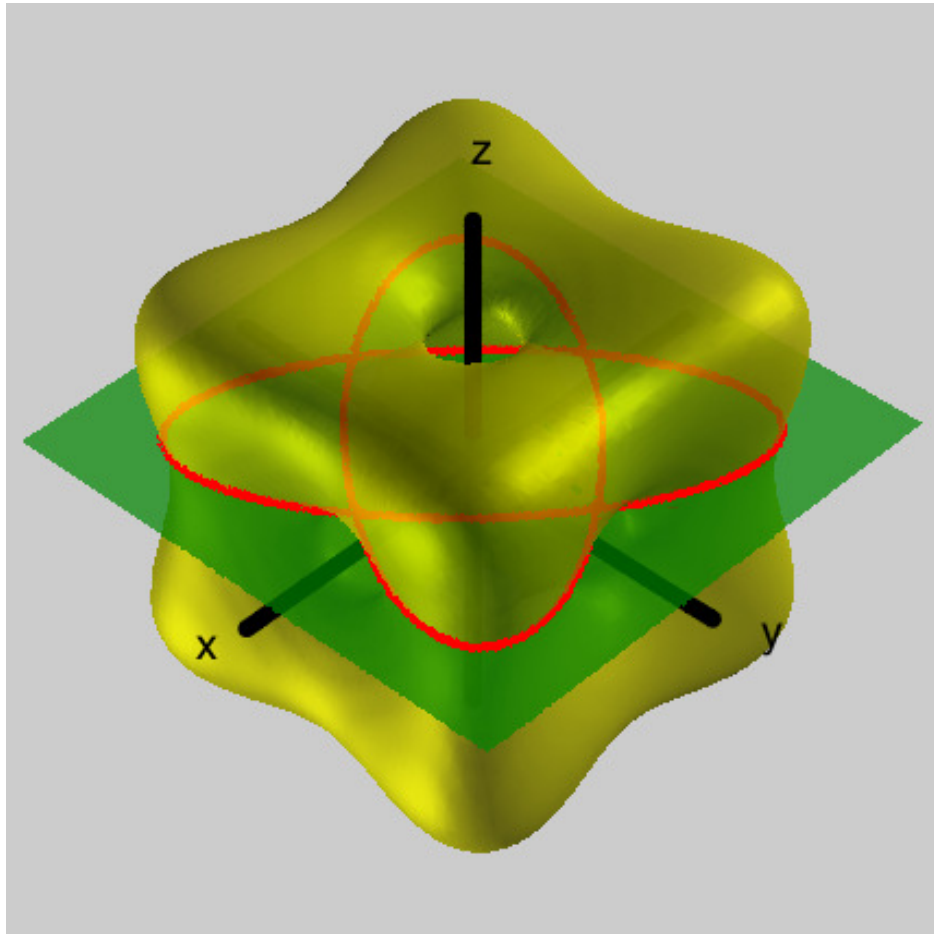
Although work exists on extracting topological features from multi-fields, an effective visualization system to support and communicate the mathematical study is yet challenging to develop. Indeed, the topological interactions and features are more complex than in the scalar case: the smallest yet interesting case maps 3-manifolds to 2-manifolds.

This chapter therefore reports a user interface (Figure 4.1) developed in collaboration with a leading expert in fiber topology. Here, rather than deploying novel techniques for general-purpose data, the primary focus is on supporting existing mathematical workflow and conventions, with departures from these conventions deliberately limited. The user interface visualizes the complex topology of fibers for the class of functions  $f : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ . Interestingly, the *Reeb space* analysis in Chapter 3 provides necessary information. Additionally, we visualize the Reeb space in 3D, by a newly developed layout algorithm.

In the remainder of this chapter, Section 4.1 characterizes the mathematical workflow in singular fiber theory, and identifies bottlenecks where machine visualization can accelerate tasks. Section 4.2 discusses the design of the application, and Section 4.3 the new techniques required. Finally, Section 4.4 gives some implementation details, Section 4.5 discusses the use of the interface.

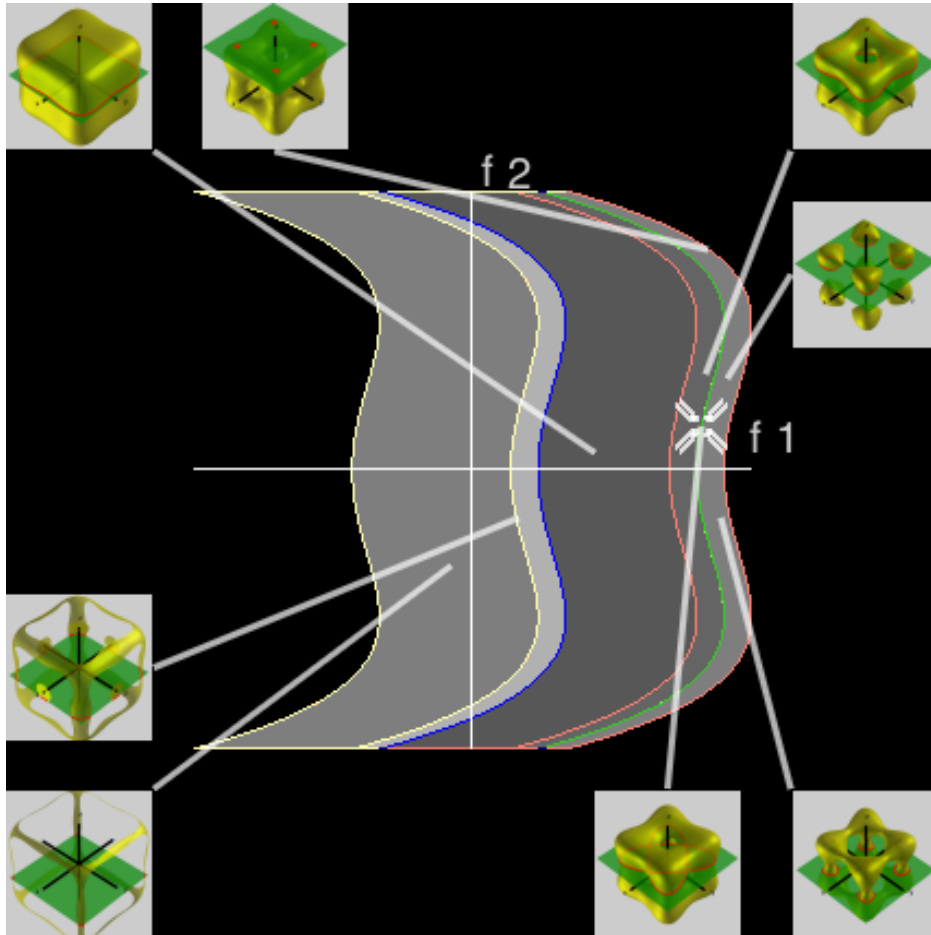
## 4.1 Tasks for Studying Singular Fibers

With the mathematical preliminaries given already in Chapter 3, the question is on how to support mathematical development of singular fiber theory. As with any application, this requires understanding the workflow into which



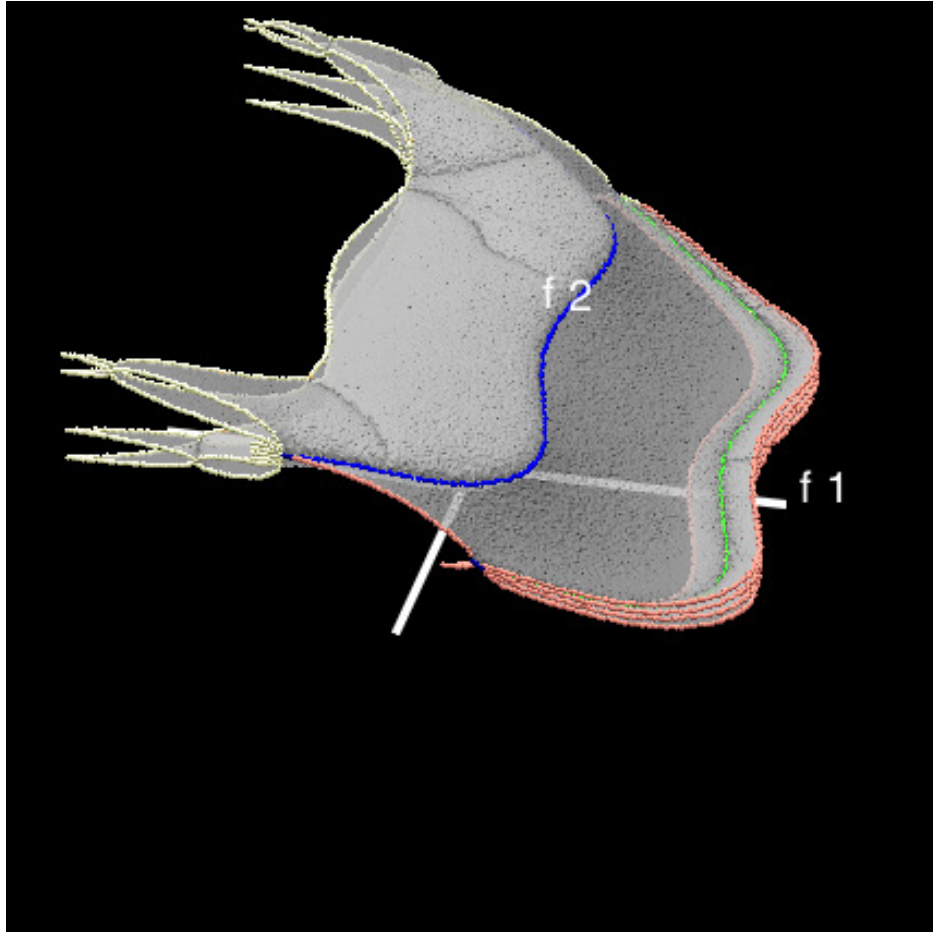
(a)

Figure 4.1: Singular fibers of the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . (a) The domain space view, with isosurfaces of individual axis function shown in yellow and green with their intersection along the current fiber shown in red. Here, the fiber is actually a singular fiber.



(b)

Figure 4.1: Singular fibers of the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . (b) The range space view, with a cross mark at the function value defining the current fiber. Critical function values are shown in different colors according to the topological types of their fibers. Arranged around this window are thumbnails of the domain space that pop up when a value is selected.



(c)

Figure 4.1: Singular fibers of the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . (c) The Reeb space window shows the connectivity of connected components in the fiber, constructed by using a 3D layout of the joint contour net.

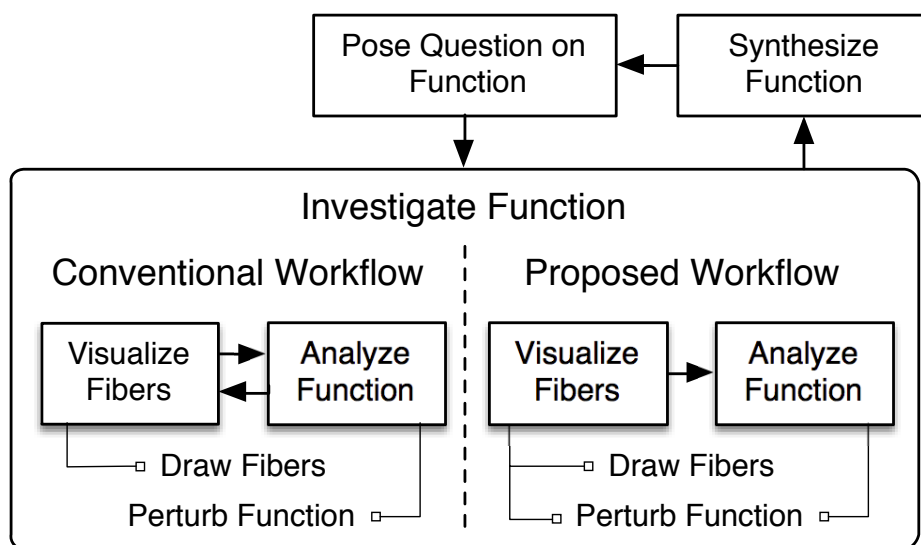


Figure 4.2: A typical workflow for studying singular fibers using manual visualization. A mathematician poses a question on a function, then investigates the function by manual analysis and sketching, perturbing the function if degeneracies are found. The investigation continues until the function has been explored, then a new function is selected, often by modifying the previous function. The process then iterates.

the application will fit, as shown in Figure 4.2.

Characteristically, this workflow starts when a mathematician poses a question about the topological structure of a function, then investigates the function using manual visualization. As shown in Figure 4.2, this starts with synthesizing a suitable function, and posing a question about its structure.

After that, the mathematician chooses some fibers of interest and sketches them in space. Viewing these fibers then gives insight into the topological changes in the function, and allows selection of new fibers. These are in turn sketched, and the mathematician gradually fills in a sketch similar to that shown in Figure 3.2, until they are satisfied that they have adequately explored the function's topology.

A new function is then selected and the process starts over, gradually building up an understanding of the complete set of possible topological behavior. In practice, however, new functions are chosen by modifying an existing function slightly, either by perturbation or in some other fashion.

While this process is well understood, it has several difficulties. First, without a great deal of experience, manual visualization of the topology is laborious. Second, the functions chosen are normally polynomials, as these are simpler to analyze. This leads to exploration of a subset of the full space of possible functions, since not all functions are polynomial, and polynomials over degree four cannot be solved in closed form. Third, the functions chosen are characteristically analytical and have relatively few topological features, both due to their polynomial definitions and the need to restrict the complexity: in comparison, data-driven problems may have tens of thousands of topological features.

As a result, computational tools for analysis and visualization have the potential to accelerate the process of mathematical discovery, and to supplement existing manual visualizations with additional information not currently exploited.

In particular, the tasks that need to be supported are:

- Drawing the fibers
- Perturbing the function

In the following subsections, we explain these two tasks and how computational visualization can help.

### 4.1.1 Drawing the Fibers

As shown in Figure 3.2, manual visualization of topological shapes is based on plotting the characteristics of the fibers in the range of the function, to help understand the configuration of fibers. In addition, isosurfaces of the two scalar functions are sometimes drawn, in order to get insight into fiber changes with respect to the function values.

These steps, however, are non-trivial, as they require solving for roots of a polynomial function. Moreover, by conceptualizing and visualizing single points in the function space, the mathematician is in effect sampling the space. Thus, while the topological shape of a fiber may be apparent, the relationship with nearby fibers depends on understanding which loops or components map to other components.

In short, this analysis depends on understanding the Reeb space [Lev85, EHP08]. Although singular fiber theorists have considered the concept [HS13], sketching the topology has not been a common strategy when understanding a function. Part of the reason for this is that the Reeb space is itself difficult to visualize and analyze, especially for complex data. Since recent work has enabled computational approximation of the Reeb space [CD14], we have added to the application the ability to visualize the Reeb space directly, as well as its relationship with the conventional range diagram.



### 4.1.2 Perturbing the Function

As noted above, singular fiber theory, like Morse theory, is most straightforward if no degenerate fiber components exist, and this is usually handled by adding small perturbations to the function to destroy the degeneracies. In practice, this occurs frequently since polynomials with complex behavior are relatively easy to define, but polynomials with multiple roots commonly have degenerate fibers. As a result, the first iteration of the workflow commonly starts with a polynomial with degenerate fibers which is progressively edited until the degeneracies are eliminated.

Given that the functions studied are usually polynomials, the normal strategy for perturbation is to add another polynomial to the original function, so that the new function is slightly different from the original one. Unfortunately, it is hard in practice to find a suitable additive polynomial that destroys degeneracies but does not otherwise alter the function significantly. As a result, degeneracies are difficult to eliminate systematically, and a considerable amount of trial and error is required in practice.

Thus, one of the key features of any application for this branch of mathematics is the ability to apply perturbations to the data and show the resulting topology immediately, with the option of removing the perturbation and trying again.

## 4.2 The User Interface

As we have seen in the previous sections, while mathematical development of singular fiber theory is well advanced, it is strongly dependent on manual visualization of functions, and these manual visualizations are the bottleneck in terms of time. This motivates the development of visualization interfaces which accelerate and improve this process, but these interfaces must support the workflow of the mathematician, and in particular the iterative

examination and perturbation of functions, using the existing conventions of range-based visualizations.

The principal design decision, therefore, is that at least one view must show the conventional range diagram: i.e. a 2-D rendering of the fibers and the boundaries of topological change. Secondly, since interpreting this diagram relies on being able to understand the relationship between the fibers and the function, a view is required in which the fibers are drawn in the domain. And thirdly, since comprehension of the Reeb space is at the heart of the mathematicians' task, a third view should show the Reeb space. We will call these views the *range view*, the *domain view* and the *Reeb space view*, respectively, as shown in Figure 4.3.

As shown in Figure 4.3 again, other supporting GUI components are available for interacting with these views. The user can start/end perturbing the function by checking/un-checking the check box on the left. The user defines the function by inputting equation in the text field. The color of fibers can be changed by clicking buttons and specifying colors with an operation system-specific color picker. Here, the buttons represent the topological shapes of a fiber component. The shapes are categorized as they were in Figure 3.7, so that the user can understand the topological shapes of fibers by taking advantage of our fiber detection algorithm. The user can also perform screen captures of the views from the menubar.

We start with the domain view, as it is the simplest, then the range view and Reeb space view, omitting discussion of function perturbation until the basic role of each window has been covered.

### 4.2.1 The Domain View

Mathematical exploration of multivariate topology depends on being able to see the topology of individual fibers, and the domain view is present for this purpose. As described in Section 3.1, a fiber may conveniently be described

as the intersection of isosurfaces of two different functions, and this approach is commonly used already.

The first view, therefore, shows a single fiber by displaying one transparent isosurface for each of the two functions in contrasting colors, and showing their intersection in a third color. In Figure 4.3, these colors are yellow, green and red respectively, but these choices are arbitrary and other possibilities could be chosen.

As usual for renderings of 3D data, this view can be rotated, while this should primarily be understood as a passive view in which the results of choices made in the range view are displayed.

### 4.2.2 The Range View

Since mathematicians already depict fiber topology in the range space, this view is the heart of the application. Ideally, this view would be identical to the diagram shown in Figure 3.2, with boundaries between regions of different topological shape, and each region or boundary neatly labelled with an iconic representation of the topology. In practice, however, once functions become arbitrarily complex (as shown in Figure 3.4), it is easy to overload the visual representation. Moreover, if boundaries overlap each other, it can become difficult to identify which icon belongs to which boundary.

Instead, we choose to represent the topological changes along the boundaries with the color coding shown in Figure 3.3, by providing inset windows showing the fibers at a particular pixel, and by allowing the user to select a pixel to be shown in detail in the domain view. Thus, by interacting with the range view, the user can understand topological changes directly, by seeing how the fiber topology changes across a boundary directly.

Secondly, while color coding is generally to be avoided unless there are only a few distinct colors, in this case, there are exactly seven possible topological changes (Figure 3.3), making color coding a viable option. Moreover,

the topological change across a boundary is consistent along the length of the boundary, meaning that a uniformly colored line gives reliable information to the user.

Once the choice has been made to represent topological changes (i.e. singular fibers) with color, remaining pixels must be monochrome, but intensity can still be used to represent further information, and here we choose to use intensity to represent the *multiplicity* of the function – i.e. the number of distinct fiber components that make up a fiber. Thus, black indicates regions with no fibers because no domain point takes on those values. Dark grey corresponds to one fiber component (i.e. a simple arc or a loop), while lighter grey indicates regions of greater topological complexity.

Since it is only possible to show a single fiber at a time in the domain view, the range view also acts as a selection panel in which the parameters of the fiber are chosen, and the currently selected fiber is shown as a cross in the view. Interestingly, this view can then be seen as a projection of the graph of the function akin to that used in the work on continuous scatterplots [BW08].

While the range plot can extend to the boundaries of the view, a further refinement was added: inset views. In essence, the iconic representations in Figure 3.2 are thumbnail sketches of representative fibers, and are used as visual anchors for mathematical reasoning. Since these images are generated for the domain view, it is straightforward to compute thumbnails for any desired choice and pin them to the window for reference, as shown in Figure 4.1, using menu commands on a pop-up menu. To avoid having too much visual complexity, these are placed inside the boundary of the range window, with lines indicating their location in the range: inset placement will be discussed in Section 4.3.1.

### 4.2.3 The Reeb Space View

In addition to the range view, the Reeb space view is also provided, to allow the user to see the relationships between the sheets whose overlap is displayed in the range. Although the range view can be seen as a simple projection of the Reeb space view, there are two reasons why it should not be subsumed into the Reeb space view. First, the range view reproduces the existing mathematical convention for reference. Second, the range view is also used to choose fibers to display in the domain view, a task that would be needlessly complicated if there were free-form rotation.

Against this, it is clear that the Reeb space view gives additional information beyond that is represented in the range view. As with the domain view, rotation is supported, although user interaction is at present limited.

### 4.2.4 Function Perturbation

While the views so far described provide the functionality of drawing the singular fiber topology, the task of perturbing the function needs to be added, and at this point a problem arises: in which view should function perturbation (i.e. visual editing) be performed? To see why this is a problem, consider a degenerate fiber in which three loops occur, as in Figure 3.4. Eliminating one of these loops implies editing function values along the loop, i.e. changing values according to where they are in the domain. Moreover, values near the loop will also need to be changed to ensure continuity.

Logically, therefore, function perturbation belongs in the domain view, and this requires a choice of how to edit fibers. Since the goal is to shift the values on a fiber which lies along the intersection of two isosurfaces, a simple approach is to move either isosurface to destroy the fiber. This is achieved by letting the user select a control point on either isosurface and drag it along the normal of the isosurface at that point. Once the user is confident that

sufficient perturbation has been performed, the function values are updated as described in Section 4.3.3 and the topology of the function is recomputed and redrawn.

## 4.3 Techniques

In order to support the mathematical workflow, we would ideally compute the Reeb space of the function together with all of the singular fibers. Then, a fiber component is equivalent to a point in the Reeb space. Unfortunately, no practical algorithm has yet been reported for correct computation of the Reeb space, although an algorithm was reported for computing the Jacobi set [EH04].

Instead of this, we therefore approximate the topology of fibers by computing the Joint Contour Net (JCN) [CD14], as discussed in Section 3.2.1. Although the JCN can be used to approximate the Reeb space, it does not identify the singular fibers, nor classify their topological shapes. We therefore implement an approximate classification based on local relationships between nodes in the JCN, as described in Section 3.2.2.

Other tasks required include placement of the insets in the range view (Section 4.3.1), rendering the isosurfaces and fibers in the domain view (Section 4.3.2), editing the function in the domain view to remove degeneracies (Section 4.3.3), and laying the JCN out in the Reeb space view (Section 4.3.4).

### 4.3.1 Placing Inset Views

We saw in Section 4.2.2 the need for insets of fibers in the range view. As the session continues, however, multiple insets accumulate, and it is necessary to keep them organized. We reserve the outer border of the range view for insets, and use the center for the actual range. Then, where should the insets

be placed for maximum clarity?

We note that insets are just labels, reducing the problem to a question of label placement. For this, we apply a standard technique [BKSW07], which arranges large annotation labels outside a central content area with minimal overlap.

Once these insets have been placed, a leader line is drawn from each inset to its range value. Since the leader lines are hard to distinguish from singular fiber curves, we show the insets with an optional black boundary, as seen in Figure 4.1.

### 4.3.2 Extracting Isosurfaces and Fibers

Rendering the domain view is straightforward, as what is required is to draw isosurfaces of the two scalar functions, plus the fiber at their intersection. The latter is extracted by extracting the level set of the  $f_2$  from the patches of marching tetrahedra for  $f_1$ .

### 4.3.3 Perturbing the Function

As we have seen, one of the key tasks in visualizing singular fibers is to perturb the existing function to remove degeneracies. Doing so requires identifying fibers where multiple topological changes occur simultaneously, and changing the function slightly so that these changes are spread out over multiple fibers. In practice, what this implies is choosing one fiber component in the domain view, and modifying it so that any change to it occurs at different function values.

For this, we observe that the change must be localized in space so as not to affect the other fiber components in the fiber. While it is theoretically possible to select a fiber and adjust neighboring values automatically, a simpler approach adds or subtracts a small function with limited spatial support. If this is performed at a singular point in the domain where two components

touch, then the singular point will have its value changed, and will instead occur at another values.

Moreover, it is not necessary to displace the singular point with respect to both functions. Instead, it suffices to do so for a single function: i.e. we want to locally displace one of the isosurfaces used to define the fiber.

Once the problem is stated thus, the solution is straightforward. We allow the user to select any point on either isosurface and drag it inwards or outwards (i.e. constrained to the isosurface’s normal vector). Dragging it outwards expands the isosurface (adds value), while dragging it inwards contracts the isosurface (subtracts value).

In either case, the change is accomplished by adding a small radial basis function (RBF), the amplitude of which is governed by the distance of the drag, and the radius of which is a small distance chosen as an initial parameter: in this implementation 50% of the size of the domain. In our experiment, the radius was large enough for avoiding unintended change to the topology.

#### **4.3.4 3D Layout of the Reeb space**

In the Reeb space view (Section 4.2.3), we show the Reeb space in 3D to allow the user to understand the relationships between fiber components. Ideally, this would involve a canonical layout algorithm for the nodes of the JCN, which represent quantized fibers. Each node has its  $(x, y)$  coordinates fixed by the two function values, leaving the  $z$  coordinate available to the layout algorithm. Unfortunately, configurations exist in which the sheets in the Reeb space are guaranteed to have intersections, making a canonical crossing-free algorithm impossible.

Since this exact problem arose in work on contour trees, and required separate work to resolve [HSCS11], we can identify that this is a subtask beyond the scope of the present thesis. Thus, like the solution used by Duke et al. [DCK<sup>+</sup>12], a heuristic layout is called for, based on a spring layout



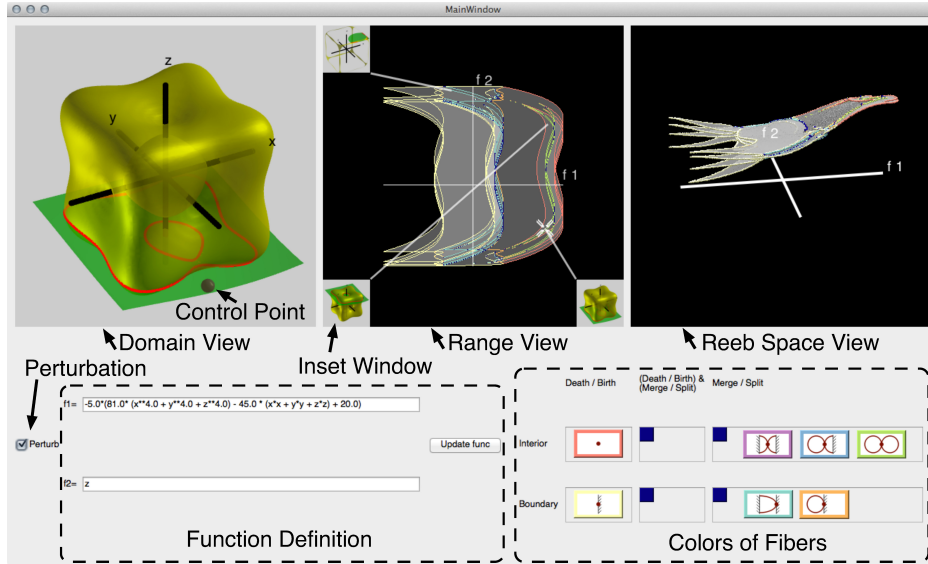


Figure 4.3: The interface consists of a set of three views, each showing a different aspect of the function under study. We put optional GUI components for interacting with these three views.

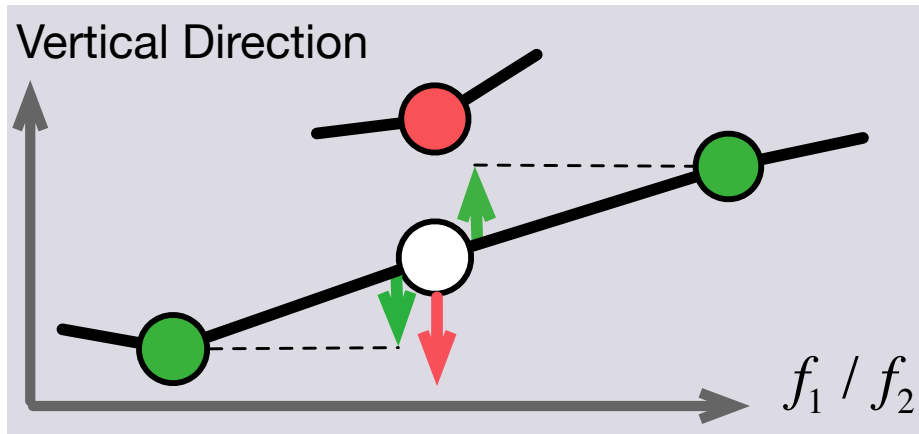


Figure 4.4: For controlling the 3D layout of the interior JCN in the Reeb space view, each node in the JCN moves vertically. The motion is controlled with the repulsive forces from other nodes having the same quantized  $(f_1, f_2)$  value (in red), and the attracting forces from the adjacent nodes (in green).

algorithm, shown in Figure 4.4. The popular force-directed graph layout algorithm [FR91] in 2D works fine for the 3D layout of the Reeb space in our experiments, but with a few modifications. We constrain the layout to leave the  $(x, y)$  coordinates fixed, and initialize all nodes greedily so that two regular or birth fiber component connected in the interior JCN with an edge have the same  $z$  value, where such a set of nodes is called a *sheet* in this thesis. The  $z$  values are determined evenly, and then normalized. We set the  $z$  value of the other nodes to the mean of those of the adjacent nodes.

To apply the forces, we set  $k_i$  for each interior JCN node  $i$ , replacing the “area” in the original equation [FR91] with 1:

$$k_i = C_i \sqrt{\frac{1}{\#\{\text{vertices having the quantized } f \text{ value } (f_1(i), f_2(i))\}}}, \quad (4.1)$$

where  $f_1(i)$  and  $f_2(i)$  denote the corresponding function value for the JCN node.  $C_i$  is experimentally set to control the layout for each internal JCN node  $i$ , as mentioned later in this Section. Then, further following the formulation in the original algorithm, the attractive force  $F_a(d) = d^2/k_i$  and the repulsive force  $F_r(d) = -k_i^2/d$  are applied as illustrated in Figure 4.4. Here,  $d$  is the distance to a node having the same quantized  $(f_1, f_2)$  value, and an adjacent node, respectively. Either  $F_a$  or  $F_r$  is calculated for each of the nodes, and summed up. Since the one-directional approaching of nodes inevitably increases  $F_r$  without a limit, we set a maximum value for its magnitude. 1 works fine in our experiment, and to balance the two forces, the maximum magnitude 1 is also set for  $F_a$ . In Equation 4.1, we set

$$C_i = \frac{\#\{\text{nodes in sheet } i\}}{\max_j(\#\{\text{nodes in sheet } j\})}, \quad (4.2)$$

where sheet  $i$  is the sheet containing node  $i$ , and  $\max_j$  is the maximum number of all of the sheets in the interior JCN. By setting  $C_i$  to this value, nodes with a small sheet can have greater repulsive force: otherwise such nodes will lie close to their neighbors having the same  $f$  value, since the

repulsive force is small for nodes in a small sheet. The nodes moves in the  $z$  direction by  $s (\min(\max(-1, \sum F_r), 1) - \min(\max(-1, \sum F_a), 1))$ , for each time step, where  $s$  is initially set to 1 and multiplied by 0.9 before every 100 iterations. In this way, the layout is first controlled to solve the global clutters, and then gradually more local clutters. The display sets an independent coordinate: it takes the  $(f_1, f_2, z)$  values as input, and the  $z$  values are normalized for each time step.

## 4.4 Implementation

In the previous sections, we described the interface and algorithmic issues for this application. We have implemented it in C++ using Qt for the interface, CGAL for tessellating the domain space, Boost for handling graph data structures, OpenGL for rendering, and again OpenGL for the rasterization step in the JCN construction. The functions were sampled at a domain resolution of  $50 \times 50 \times 50$ , although this can easily be varied.

Although performance is not the principal concern in this implementation, we give some typical figures for computation times. For every example in the Figures, the computation for the sampling, rasterization, and JCN construction ended within 2–3 minutes in total. However, the topological shape detection with the decision tree in Figure 3.7 took 15 minutes (Figure 4.1) to 2 hours (Figure 4.5, bottom). We later found that the memory layout for the JCN was inefficient, taking more than 95% of the time for access, and are working to fix the problem. These times were recorded on a laptop PC with Intel Core i7 CPU with 2 cores (2GHz, 256KB L2 cache per and 4MB L3 cache), 8GB RAM, and an Intel HD Graphics 4000 GPU (1024 VRAM).

We report that errors in fiber type detection can be found: this however was expected given the quantization implicit in the JCN. Additional errors might result from calculation errors in the function values, creating small

singularities in the domain. In practice, however, these merely change the local structure of the JCN, and that of the Reeb space visualization, without a significant impact on the global structure. Even such errors were topologically consistent: e.g. simultaneous birth and split of a fiber in a pixel were sometimes separated to a sequence of birth and split fibers in neighboring 2 pixels. In the following experiments, the errors did not affect the users' task.

## 4.5 Outcomes

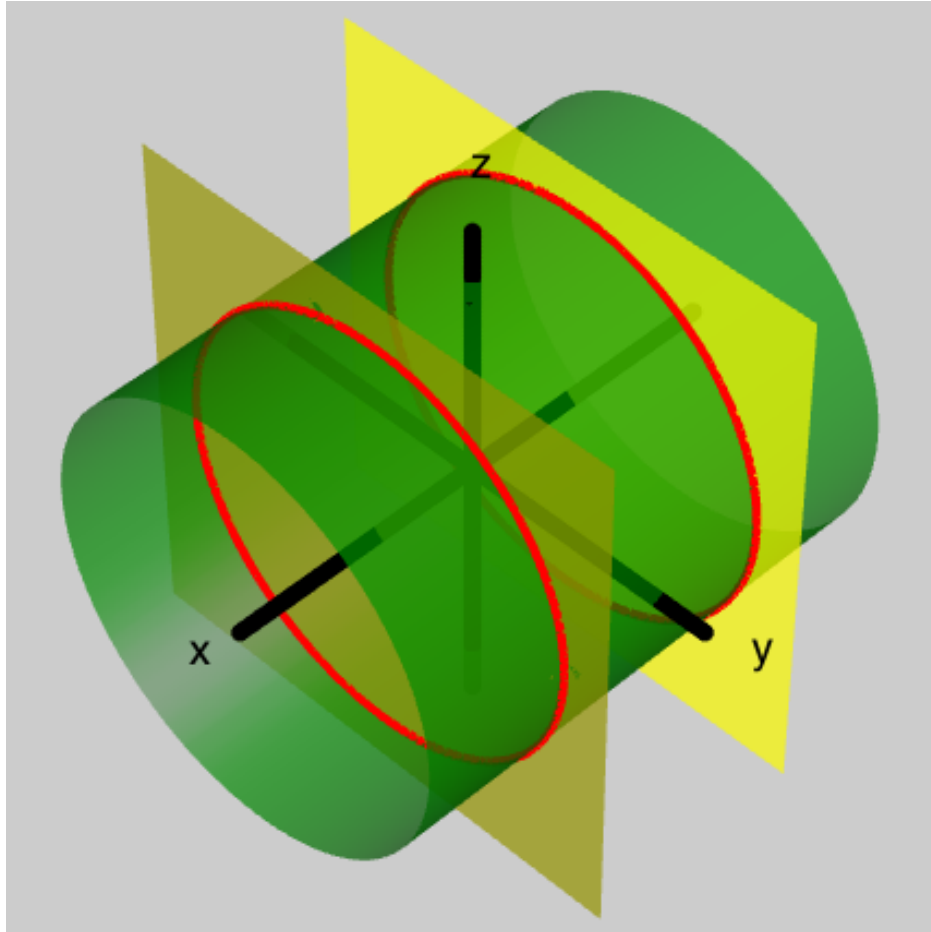
More important than the computational speed is the question of whether the interface supports the desired interactions by the user. We assess this by illustrating how the application supported a user's tasks in Section 4.5.1, and by reporting feedback provided by some indicative users in Section 4.5.2.

It is the nature of this type of application that the initial target audience is small and highly specialized – i.e. audience including a specific research group headed by one of the collaborators. Moreover, although tasks have been defined in Section 4.1, these tasks are hard to measure quantitatively, and it is more important to understand whether the target group feels that the application is successful in supporting their research. We therefore recruited two graduate students and 4 experienced researchers. Of the latter, 2 were collaborators, 1 a member in a collaborator's group. We demonstrated the application to them, and collected their feedback when done.

### 4.5.1 Task Support

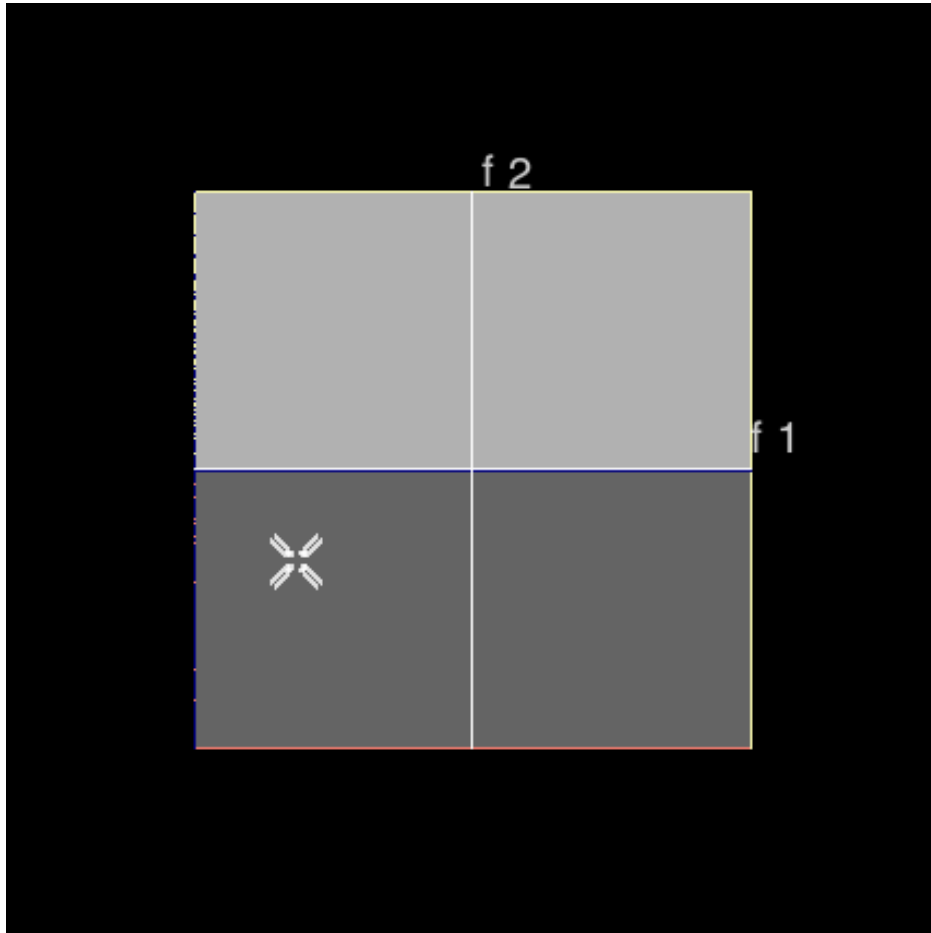
To demonstrate that the application supports the desired tasks, we focus on the two principal tasks from Section 4.1: drawing the fibers, and perturbing the function.

Figure 4.5 demonstrates how a professional mathematician (not a collaborator) used the interface for understanding the fibers in a particular func-



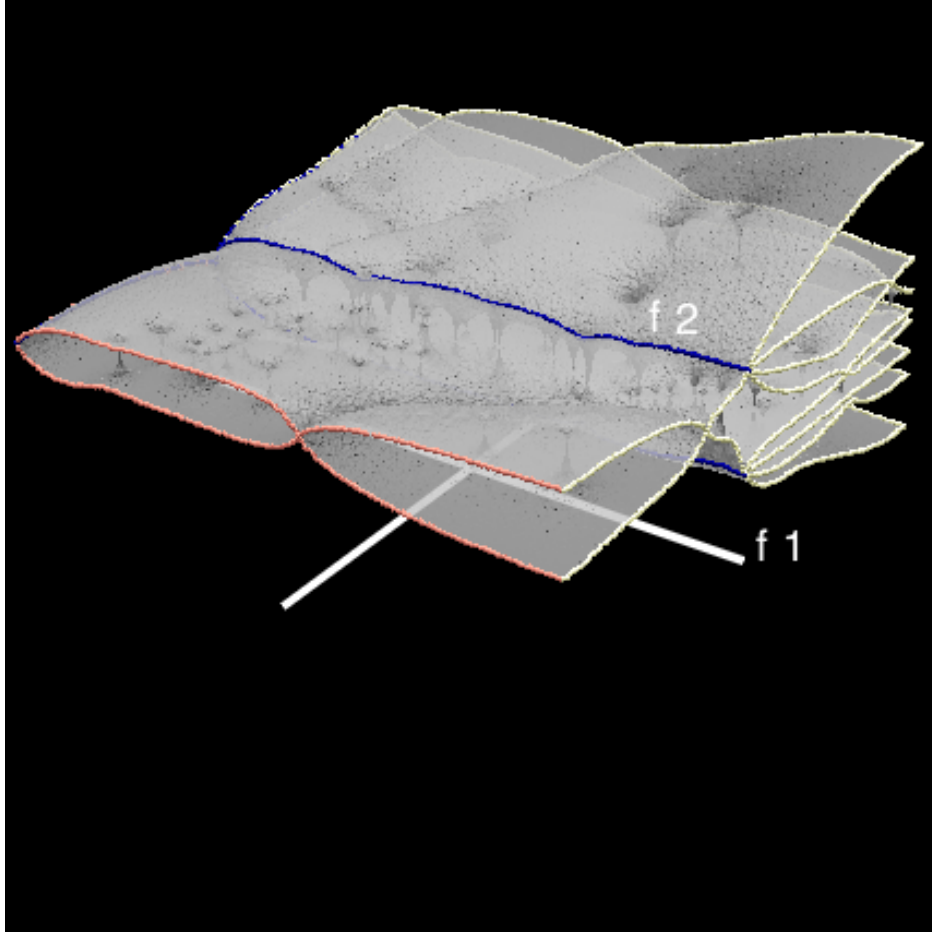
(a)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (a) The original function  $f(x, y, z) = (x^2, y^2 + z^2)$  in the domain view.



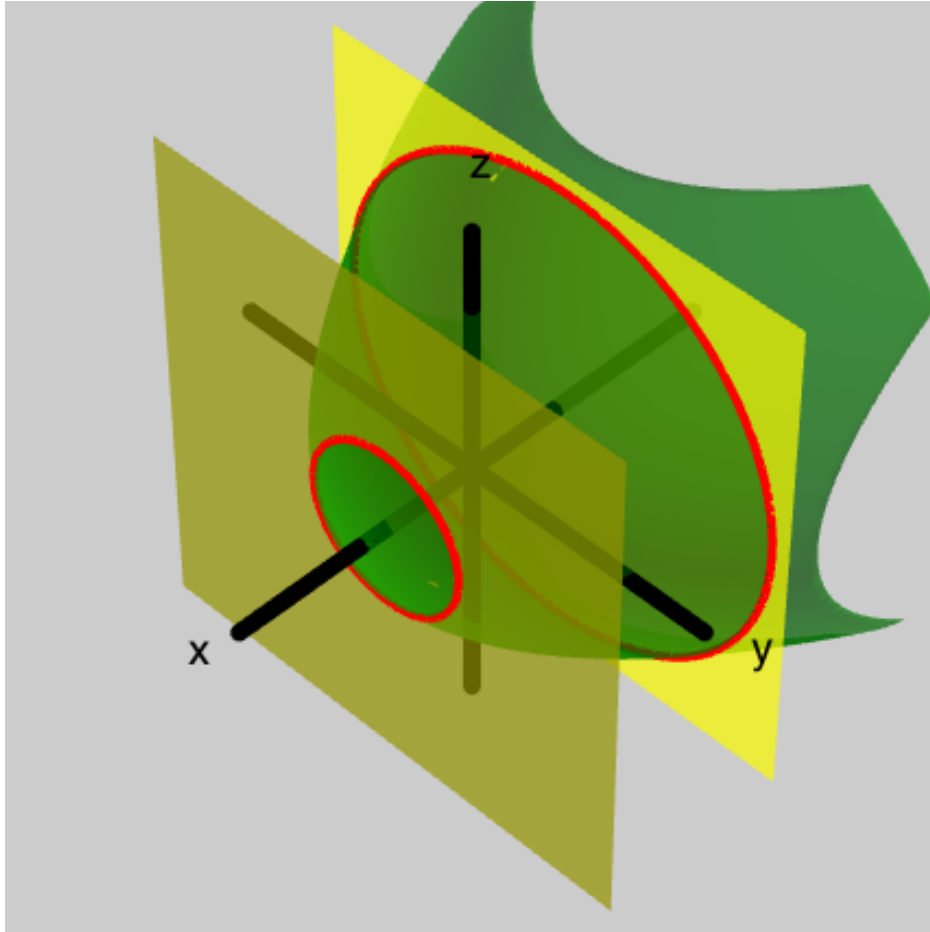
(b)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (b) The original function  $f(x, y, z) = (x^2, y^2 + z^2)$  in the range view.



(c)

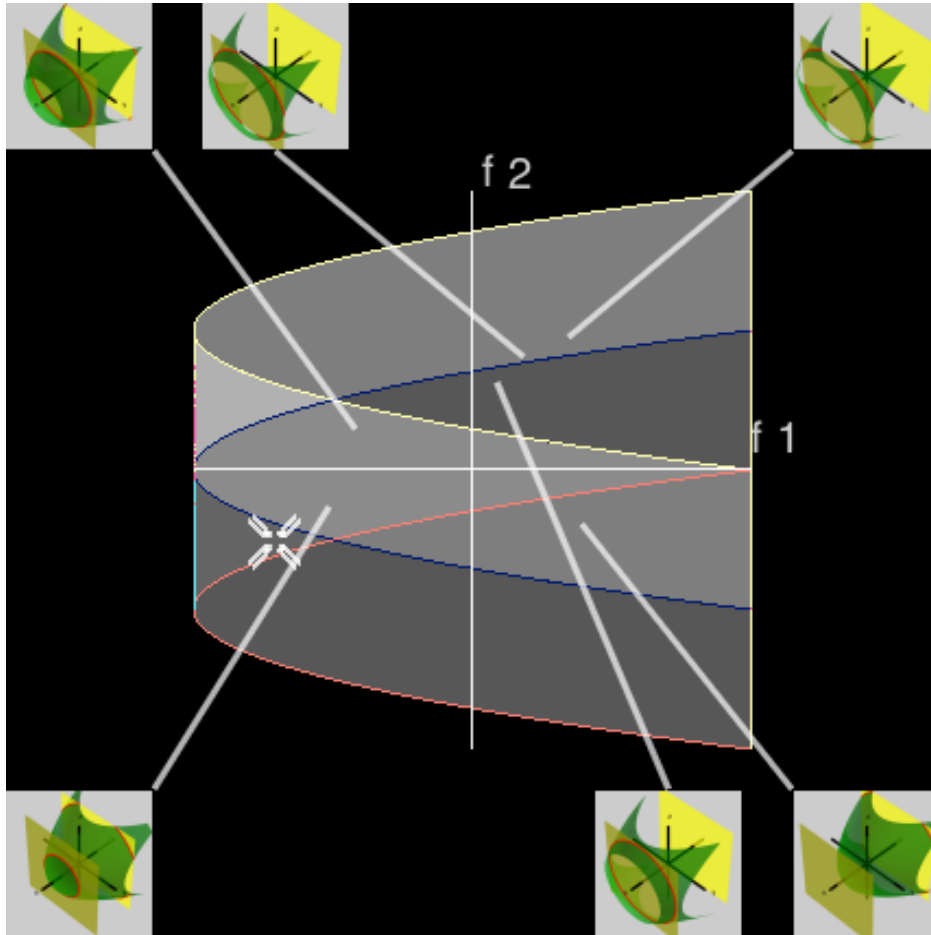
Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (c) The original function  $f(x, y, z) = (x^2, y^2 + z^2)$  in the Reeb space view.



(d)

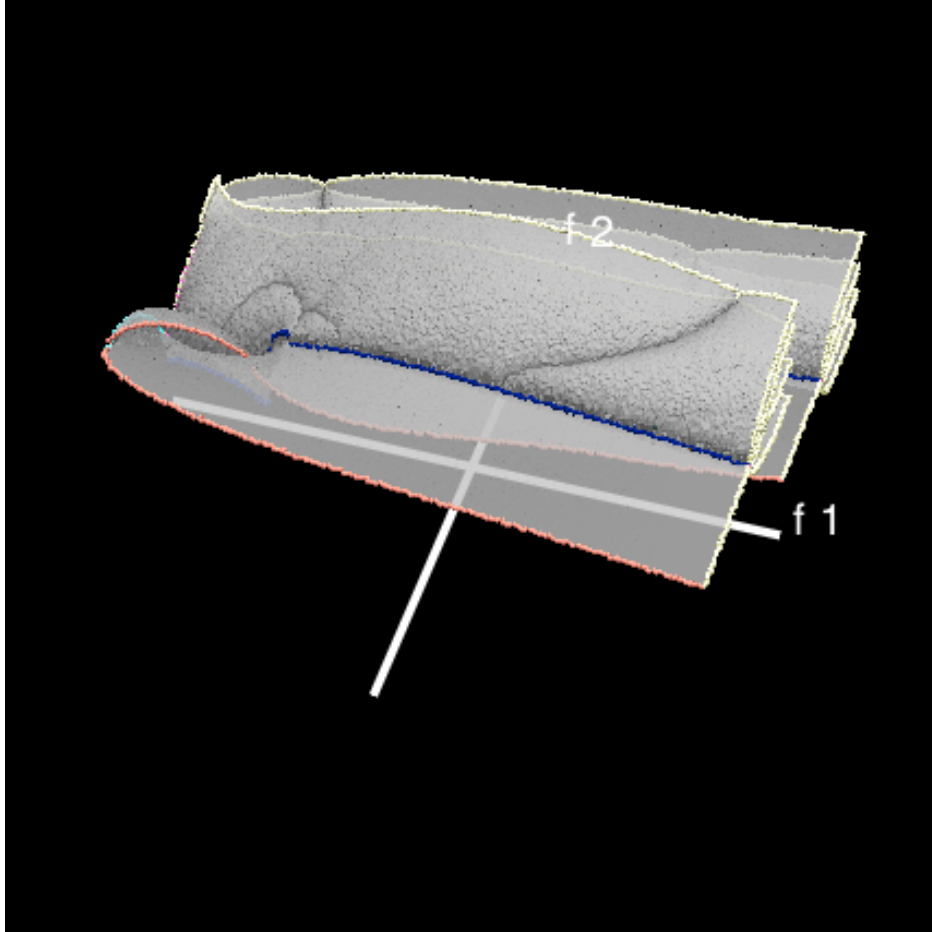
Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (d) The perturbed function  $f(x, y, z) = (x^2, y^2 + z^2 + x)$  in the domain view.





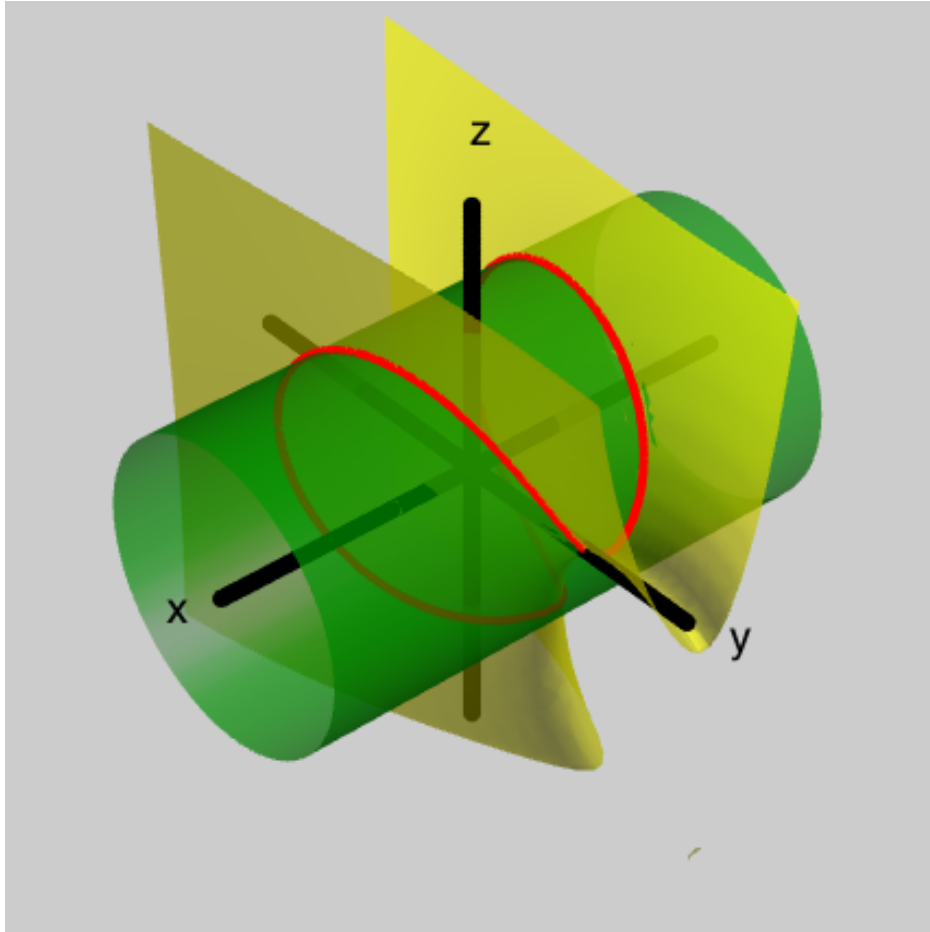
(e)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (e) The perturbed function  $f(x, y, z) = (x^2, y^2 + z^2 + x)$  in the range view.



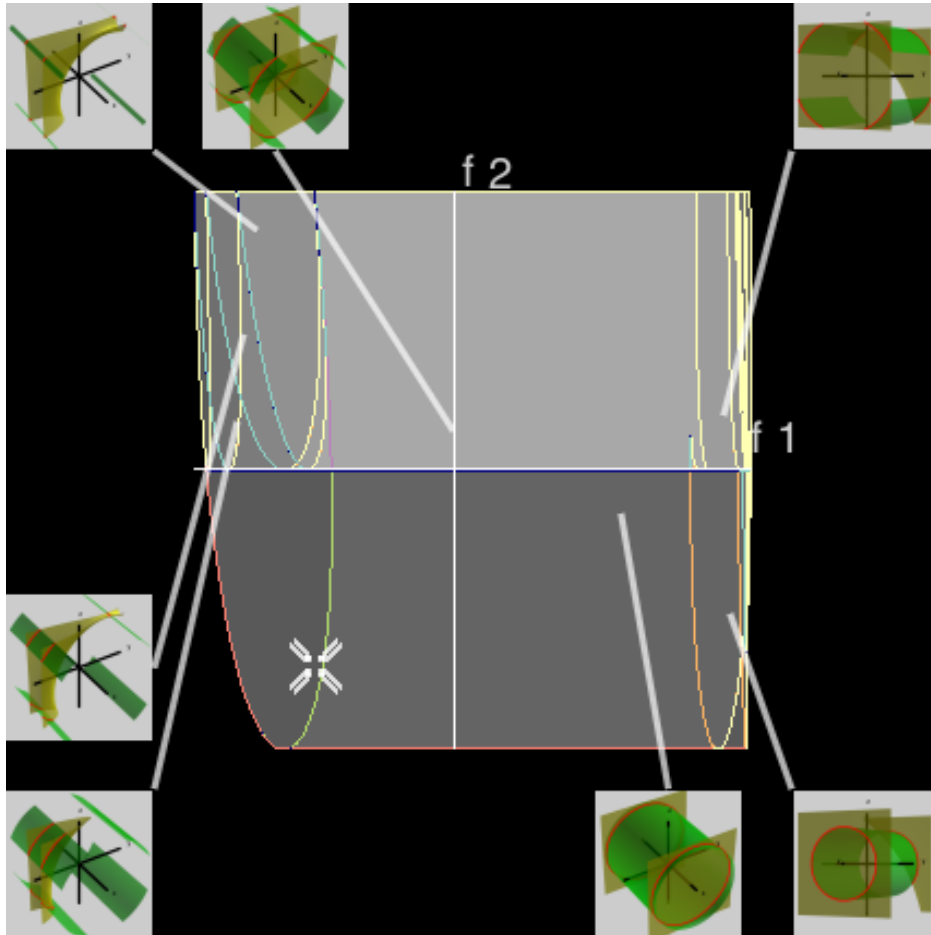
(f)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (f) The perturbed function  $f(x, y, z) = (x^2, y^2 + z^2 + x)$  in the Reeb space view.



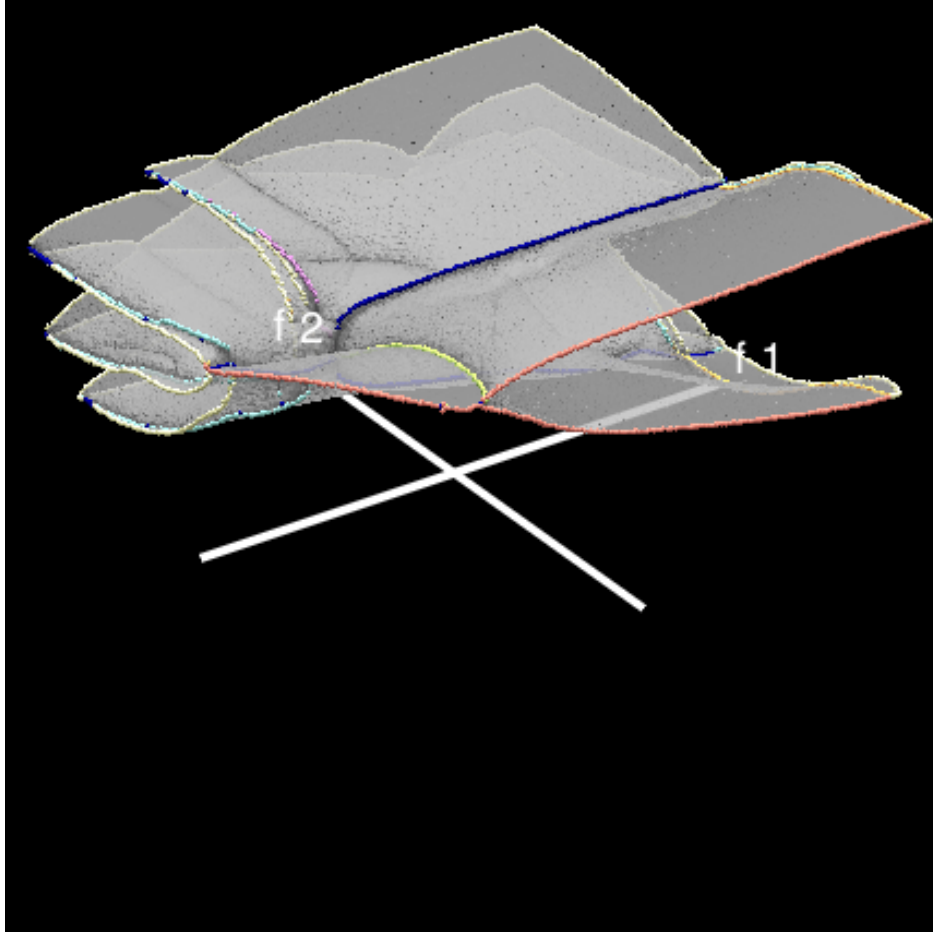
(g)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (g) The function perturbed with the RBF interface, displayed in the domain view.



(h)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (h) The function perturbed with the RBF interface, displayed in the range view.



(i)

Figure 4.5: The expert in Section 4.5.1 used the interface to investigate how a perturbation can alter the function on top,  $f(x, y, z) = (x^2, y^2 + z^2)$ . Firstly, he changed the equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ , on the middle, by applying perturbation, which is the traditional technique in the field. He ended up deforming the isosurface directly with the RBF interface, as shown on the bottom. (i) The function perturbed with the RBF interface, displayed in the Reeb space view.

tion:  $f(x, y, z) = (x^2, y^2 + z^2)$ . Here, the cross-mark in the range view defines the isosurfaces and fiber to be shown. As the user drags the cross-mark across colored boundaries, the fiber changed its topological shape. The interface satisfied the user for investigating the fibers of  $f(x, y, z) = (x^2, y^2 + z^2)$ , by helping him find a few topological deformations to the fibers. Note that this would have been time-consuming with manual visualization. After understanding the original function, he deformed it to investigate the possible deformations. He first used the traditional technique: perturbing the function with equation. He altered the input equation to  $f(x, y, z) = (x^2, y^2 + z^2 + x)$ . Then, he moved the cross-mark around in the range to see and understand the configurations of the fibers in the domain.

After that, he interacted with the RBF perturbation. In contrast to the traditional perturbation on equations, the deformation immediately showed how the fibers can be modified. He reported such a visual indication is indeed useful for understanding the various possible deformation, which had been difficult to do manually. He also reported that the 3D Reeb space view was useful for understanding the connectivity of the fiber components, compared with the Range view.

The user also set inset views to remind himself of the topology. He reported that interactions in the domain view with other two views is significantly more intuitive, compared with traditional techniques.

### 4.5.2 Informal User Study

All participants agreed that the interface successfully provided an intuitive means for analyzing the singular fibers.

The graduate students commented that the domain was useful in elucidating cases where the fibers were knotted, and for understanding complicated merging/splitting of fiber components. An experienced researcher reported that drawing the isosurfaces of the coordinate functions  $f_1$  and  $f_2$  enabled

him to investigate the relationships between the two functions. For an example of such relationships, he referred to whether the combination  $f = (f_1, f_2)$  would become a stable map. Here, a stable map is a map that contains only non-degenerate fibers. The experienced researchers felt that the rapid display of the overall configuration of the range was useful in understanding the global features of a function. This is especially useful since the interface can show how the local features, such as topological shapes, relate to each other. In particular, such relations included the number of fiber components. The experienced researcher in Section 4.5.1 and the primary researcher reported that the Reeb space view was useful for understanding the connectivity of the fiber components. However, when the Reeb space included visual clutter due to its complexity, the participants preferred the range window to identify the connectivity of the fiber components. A particular problem was that the 2-manifolds in the Reeb space were sometimes located too close to each other, which makes the structure of the Reeb space difficult for the participant to see.

Solving degeneracies also got attention of the participants. The graduate students reported that our interface let them imagine various ways of solving a degeneracy. In contrast, the principal researcher supported the use of the interface for the following reasons. First, it simplifies understanding the topological shape of a degenerate fiber by showing the decomposed non-degenerate singular fibers. Secondly, generating specific examples for illustrations becomes considerably easier, which is a classic visualization task. Thirdly, he expects the interface to be useful in studying *singularity theory*, where it is often important to guess how a perturbation changes the singular points.

The participants also felt that while the interface was useful, they wanted us to enhance the usability. In particular, improved performance was essential for the interface to become practical. Also, the participants found

that the interaction in the RBF perturbation should become more intuitive. The participants further demanded the specification of function values in the range view, instead of mouse click in the window. Indeed, an experienced researcher reported that specifying a pixel with a mouse was a difficult task. Notwithstanding this, they felt that the application would reduce the difficulty of their study. We are currently making our efforts to solve the above mentioned problems.

In addition, we tested our interface for mathematical studies in [STS<sup>+</sup>14]. A prototype of our interface had visually verified the theoretical prediction [IS09] about degeneracy in a mapping under the study. (The result is shown in the work by Saeki et al. [STS<sup>+</sup>14, Figure 17]) This shows that our interface can visually check the soundness of certain mathematical predictions and results. Although rigorous verifications are still necessary, visual checking reduces the amount of such a task in certain situations. For example, when the predictions or results are inconsistent with the visualization results, a mathematician can focus on the investigation of this inconsistency to accelerate the verification process. Further, the experienced researcher in the task support demonstration (Section 4.5.1) reported that the interface had a potential to lead to a new direction of mathematical theories by providing visual instincts that were difficult to obtain before.

In summary, both experienced and relatively novice participants found the interface of value, albeit in different ways. Since the intent was to support both analytic and learning tasks, the feedback confirmed the application's value of the application, while identifying directions for further development.



# Chapter 5

## Simplifying the Time-Evolution of Local Extrema

In this chapter, we simplify the time-evolution of local extrema in scalar fields. Existing simplification techniques have not taken into account the temporal continuity of local extrema, as reviewed in Section 2.3. In contrast, we obtain temporal continuity by analyzing fiber topology. This is based on our Reeb space analysis in Chapter 3. Evaluation is given by analyzing fluid simulation datasets. Before proceeding, however, recall that the preceding chapters focused on the analysis of multi-fields, but not time-varying scalar fields. Therefore, first we explain our contributions in the context of analyzing the time evolution of local extrema.

### 5.1 Analyzing the Time-Evolution of Local Extrema

The time-evolution of local extrema is analyzed in a variety of applications [LHL<sup>+</sup>98, LBM<sup>+</sup>06, TFTH13]. For example, in the study of magnetic fusion plasma, flows of ions and electrons are analyzed as 2D distribution of electrostatic potential [LHL<sup>+</sup>98]. There, local extrema correspond to vortices. The vortices are found in both macroscopic and microscopic scales, and their

complex interactions are studied theoretically and experimentally [KKSA10].

Speculations have remained about physical processes in fluids. In case of magnetic fusion plasma, for instance, zonal flow structures can be formed [LHL<sup>+</sup>98]. Physicists have described the phenomena with a 2D turbulence model [HM78], but the details of the structure formation processes are still unclear. For a further illustration, consider a 2D fluid with anisotropic vortices that are lengthened along the vertical direction. One hypothesis may be the elimination of small vortices along that direction, while another may be the mergers of such small vortices along the same direction.

When a physicist intends to understand the time-evolution of local extrema, a straightforward and common approach is to animate static visual metaphors, such as isocontours or colormaps. However, temporal continuity of a local extremum is interpreted by human eyes, and therefore analyses tend to focus on features that stand out. Such analyses can be easily misled by a naive configuration of visualization parameters such as the number of contour lines or the choice of a colormap. Alternatively, one would understand the time-evolution of local extrema by visualizing the Fourier spectra or the autocorrelation of the time-varying scalar field. The problem here, however, is that such analyses do not provide the spatial locations of local extrema in the domain, and thus neither the locations of the vortices. As a consequence, the positions of vortices remain unknown, let alone events such as birth/death/merge/split events [SW97, LBM<sup>+</sup>06] of vortices. Therefore, sophisticated visualization techniques are necessary.

## 5.2 Objectives

As mentioned above, the time-evolution of local extrema awaits further analysis with a visualization technique that respects the temporal continuities of the local extrema. In order to cope with the vast amount of local extrema, it

is necessary to focus on the informative parts of a time-evolution. We refer to this act of focusing as *simplification*. When conducting simplification, the temporal continuities of local extrema should be taken into account, in order to understand the dynamics of fluids. Therefore, such continuities should be explicitly extracted and encoded in the feature extraction process. When visualizing the dynamics of fluids, interactions of vortices are often classified into birth, death, merge, and split events [SW97, LBM<sup>+</sup>06]. Indeed, the phenomena of plasma in the previous section can be understood as mergers of vortices, in certain cases [KKSA10]. Interestingly, it is often assumed that such temporal events of vortices are identical to those of contours. This motivates us to analyze the topology of isocontours in time-varying scalar fields.

However, dealing with critical points in realistic datasets is not straightforward. Indeed, techniques for simplifying critical points that continue over time are not fully developed. Our idea is to utilize our formulation in Chapter 3, which was based on fiber topology of function  $f = (f_1, f_2)$ . When extracting the time-evolution of critical points from a time-varying 2D scalar field  $f_1(x, y, t) : R^2 \times R \rightarrow R$ , one can define an auxiliary function  $f_2(x, y, t) = t$ . In this way, the evolution of the 2D scalar field can be expressed as  $f : R^2 \times R \rightarrow R \times R$ , which is identical to  $f : R^3 \rightarrow R^2$ . Therefore, the time-varying 2D scalar field with the auxiliary function  $f_2$  is a special case of our target function  $f : R^3 \rightarrow R^2$ . Currently, techniques are not available for exploring the local extrema that are interpolated along the time axis. Although we can find several techniques on showing and simplifying the critical points in 2D scalar fields, they cannot be directly applied to our objective since the temporal snapshots of the 2D scalar field are not interpolated over time. Therefore the simplification process has been applied individually for each time step, without fully tracking the evolution of the critical points over time. (See Section 5.3 for a comprehensive comparison

with existing techniques.) To summarize, the existing methods for tracking the time-evolution of critical points did not fully retain the temporal continuation of such points. Such methods limit the analysis of the time-evolution of critical points, in the sense that visualization results were not based on an actual interpolation.

The goal of this chapter is to simplify the time-evolution of local extrema in 2D scalar fields, and to visualize the results. Regarding this chapter, the contribution is twofold. The first is to extract the time-evolution of local extrema, which encodes the temporal continuities of them. The second is to simplify their temporal transitions by cutting off local temporal trajectories of critical points. We directly visualize the time-evolution of critical points. As mentioned in Section 1.2, identifying the local extrema from the structure of the JCN is a challenge. We overcome this problem by replacing the data values with the quantized values in the JCN, and extract the local extrema from these values. The time-evolution of local extrema is simplified with the recently developed Reeb skeleton [CCDGb], which describes the global structure of Joint Contour Net by grouping similar JCN nodes. By eliminating these groups, we can eliminate corresponding local extrema in a topologically consistent manner.

The remainder of this chapter is organized as follows. Section 5.3 clarifies the relation of our technique with existing ones. In Section 5.4, we introduce our mathematical formulation. There, we make clear the relationship between the time-varying scalar fields and local extrema. We also review the types of topological reductions that are applicable to time-varying 2D scalar field. In Section 5.5, we introduce how we extract and simplify the Jacobi set. Finally, we evaluate our method in Section 5.6.

## 5.3 Visualization Techniques for Tracking the Time-Evolution of Local Extrema

Chapter 2 included a review for analyzing time-varying scalar fields, in the context of topological analysis. Now, our contribution is clarified from the viewpoint of visualizing the time-evolution of local extrema by introducing the background. Therefore, we next review related techniques from the visualization for the time-evolution of local extrema.

In the literature, we find two major categories of techniques for extracting the time-evolution of local extrema. The first extracts local extrema for each time step separately, while the second extracts the time-evolution of local extrema as continua such as curves.

First, we compare our approach with the first category, which is techniques extracting local extrema for each time step. Bremer et al. [BBD<sup>+</sup>07] encoded the critical points in the Morse-Smale complex [EHZ03, BHEP04], for each time step. By taking advantage of the Morse-Smale complexes, they simplified the configuration of critical points by removing small ridge/ravine lines and their associated critical points, for each time step. They also pointed out the fact that the time-evolution of critical points becomes the Jacobi set by considering an auxiliary function  $f_2(x, y, t) = t$ , as we did in Section 5.2. However, they did not propose a method to extract the Reeb space. Takeshima et al. developed a hybrid wind tunnel [TFTH13] as a visual analysis tool for comparing actual flow and its measurement-integrated simulation. They visualized the actual flow with a fog generator physically, and the simulation result with an image showing topological features such as the critical points and the ridge/ravine lines. For animating the results, topological features were extracted in each time step using Takahashi et al.'s method [TIS<sup>+</sup>95]. Although simplification was not mentioned by Takeshima et al., the time-evolution of critical points could potentially be simplified by sim-

plifying the surface network for each time step. Finally, the continuity of critical points was defined as the overlaps of the vortical regions.

Kettner et al. [KRS03] developed the *Safari interface* to let the user find interesting time step. They plotted the number of contours in a 2D histogram of time and the scalar value, so that the user can effectively find interesting phenomena. Later, Fujishiro et al. [FOTT08] developed T-Map, which was a time-series of pixel glyphs indicating the topological complexity of isosurfaces. This enabled the user to effectively find the time steps of interest.

In contrast to the techniques mentioned so far, we actually extract and analyze the Reeb space thanks to the JCN, in order to trace the temporal connectivity of fibers, and then simplify it. In our approach, the temporal behaviors of critical points are defined by actually interpolating the scalar values linearly over the tetrahedra that are obtained by tessellating the 3D domain (which consists of the 2D spatial and the 1D temporal domain).

Now, we compare our approach with the second category, i.e. techniques for extracting the time-evolution of local extrema as spatiotemporal trajectories. As mentioned in Chapter 2, Edelsbrunner et al. [EHMP04] extracted the time-evolution of critical points from time-varying scalar fields together with the Reeb graphs for every time step. As with our approach, the time-evolution of critical points was identified as the Jacobi set of multi-field  $f = (f_1, f_2)$ , by setting an auxiliary function  $f_2(x, y, t) = t$ . The types of topological changes were also identified. Another important advantage in the approach by Edelsbrunner et al. [EHMP04] is their Jacobi set extraction algorithm [EH04]. As opposed to our approach, this approach does not require the quantization of function values, and therefore can extract the full details of singularity. However, their algorithm does not explicitly encode the connectivity of non-singular fiber components between consecutive time steps. In contrast, our JCN based approach utilizes such information

for selecting important local extrema. Unfortunately, their visualization results were not presented, possibly due to an overwhelming amount of critical points. Indeed, the simplification of extracted features is necessary for exploring the time-evolution, in order to cope with such overwhelming amount of critical points. Sohn and Bajaj [SB06] proposed a similar approach, which connects the nodes of Reeb graphs across different time steps. The correspondence between these nodes (i.e. contours) were identified by the overlaps of the corresponding contours in neighboring time steps, rather than interpolating the connectivity of fibers. In addition, they did not track critical points, but isocontours at a scalar value specified by the user. Shafii et al. [SDHH12] tracked the time-evolution of critical points to visualize the effect of smoothing scientific data. Rather than tracking the whole fiber topology, they attached the associate types of topological changes over the entire period of time, as labels to the initial contour tree. Even in these approaches, extraction and simplification of the critical points respecting their temporal continuity remained as an untackled problem. In contrast, we can interpolate the fiber topology with respect to the time, and can simplify the time-evolution of local extrema while respecting their temporal continuity.

## 5.4 Mathematical Formulation

In this section, we explain how a scalar field can be viewed as a multi-field, and how we formulate the simplification based on fiber topology.

### 5.4.1 Time-varying Scalar Fields and Their Jacobi Set

Now we explain how the time-evolution of critical points becomes the Jacobi set. Actually, a time-varying scalar field over a 2D domain,  $f_1 : R^2 \rightarrow R^1$ , can be understood as a multivariate data where each time step is represented with an isochron, i.e. an isosurface of an auxiliary function  $f_2(x, y, t) := t$ .

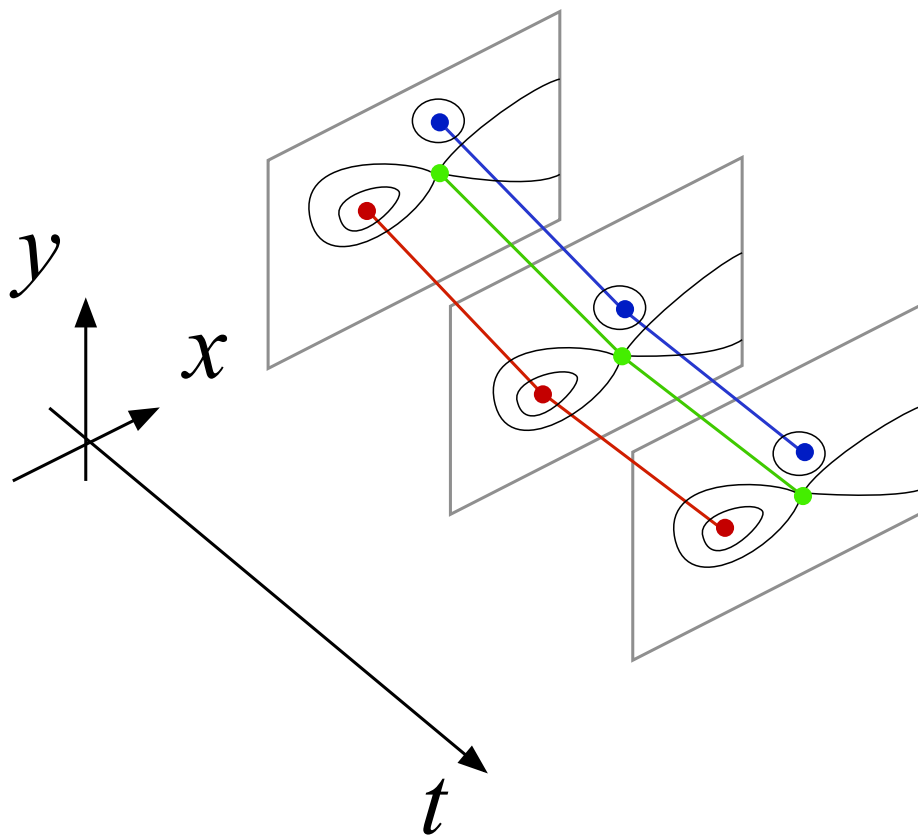


Figure 5.1: The Jacobi set in a time-varying scalar field. The local maxima, local minima, saddles are indicated in red, blue, and green, respectively.



See Figure 5.1, where the local maxima and local minima (in red and blue, respectively) are points where the fiber becomes a point, which is the birth of the fiber of  $f = (f_1, f_2)$ . Further, the saddles (in green) are points where fiber components meet to become a single fiber component. Thus, we see that the critical points are actually the Jacobi set from the viewpoint of fiber topology.

The difference to the general case of multivariate data is that one of the coordinate functions (in this case  $f_2$ ) is always a plane. One consequence is that every plane becomes one single connected component without a loop.

### 5.4.2 Simplification through Modifying Fiber Topology

Gay and Kirby [GK11] discuss how the death and birth fibers can be removed from generic smooth maps. As we have seen the correspondence between death / birth fibers to the local extrema in the above section, we can build on their analysis to address our problem of simplifying the local extrema. Figure 5.2 lists the possible topological transitions that can be used for simplifying the Jacobi set in the Reeb space, barring degeneracies.

The *swallowtail* is a pattern where a curve of critical (or singular) values will go under topological change. There, two cusp points and another fiber type appear or disappear. Bhatia et al. [BWN<sup>+</sup>13] studied this as the simplification of Jacobi set. However, implementation details were not given.

The *merge* is a pattern where two cusps are resolved to form two curves of critical values. To the best of the author's knowledge, this is not studied in the context of simplifying the Jacobi set.

The *eye* is a pattern in which a merge and a birth curve will disappear together with two cusps connecting them. Chattopadhyay et al.'s simplification through Reeb skeleton [CCDGb] is similar to this, although the connection to the mathematical theory was not given.

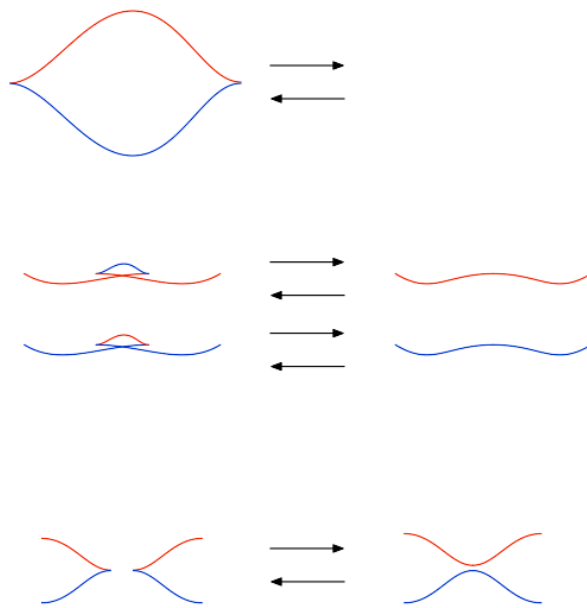


Figure 5.2: Topological changes in the range (and the Reeb space) that correspond to simplification, for a stable function. The blue curves indicate the image of merge fibers, while red curves indicate that of birth fibers.

This time, we choose eliminating the eyes for simplifying the local extrema, as it is the only one which will actually remove a connected component of critical points, rather than merely reducing the points in it.

### 5.4.3 Persistency

As with common topological simplification techniques [CSvdP04, BBD<sup>+</sup>07], we utilize *persistence* to control the complexity of the topology in the display. In this strategy, a *persistence value* is set to specify whether a feature (in this case a local extremum) is to be shown or not to the user. Here, a persistence value is a real number for each feature. In the simplest form of the simplification technique with persistence, features with persistence value less than the user-specified threshold are hidden from the visualization results, while those greater are shown. While the persistence should be calculated according to the interest of the user, we set it to the volume of the sheet, in this work. Alternatively we also tried the duration of the local extrema, but this resulted in removing many separate local extrema at the same persistence threshold. This is due to the fact that most local extrema had similar durations represented with quantized time durations, resulting in having several identical time steps.

## 5.5 Simplifying Local Extrema

On the basis of the formulation in Section 5.4, we extract and simplify the time-evolution of local extrema in 2D scalar fields. Recall that the time-evolution can be regarded as the birth-fibers in  $f = (f_1(x, y, t), f_2(x, y, t)) : \mathbf{R}^3 \rightarrow \mathbf{R}^2$ , where the domain consists of the 2D space and time. Here,  $f_1$  is the original scalar field, and  $f_2(x, y, t) = t$  the time. We obtain the topology of fibers through analyzing the Reeb space. The obtained topology is simplified by removing the eyes. The simplification is controlled by a single

persistence value: each eye has a persistence value, and it is shown if and only if its persistence value is below the user-specified threshold.

### 5.5.1 Obtaining the Connectivity of Fibers

To analyze the Reeb space, we find two approaches in the literature: Jacobi set extraction [EH04, EHMP04] and the Joint Contour Net (JCN) [CD14]. We choose the JCN for the following reasons.

On one hand, as mentioned in Chapter 2, the local extrema can be extracted as the Jacobi set [EH04] without quantization, and the *time-varying Reeb graph* encodes the temporal continuity of the local extrema with the Reeb graph for each time step [EHMP04]. To restate the conclusion in Chapter 2, however, these techniques do not offer the Reeb space. Indeed, while they provided the mathematical conditions to find the topology of fibers between two time steps [EHMP04], the actual computation for investigation was not given. As a result, the connectivity, and thus also the time-evolution, of the fiber components are not available. In particular, the image of merge-fibers can intersect between two time steps, and this information is not represented in the time-varying Reeb graph. Our simplification in Section 5.4 cannot be established without such connectivity of fibers.

On the other hand, the Reeb space can be approximated with the JCN [CD14], as we have done in Chapter 3. In contrast to the above mentioned Jacobi set extraction, the continuity of the fiber components can be obtained.

The Reeb Skeleton [CCDGB] describes the eyes. We simplify the local extrema, instead of the set of fibers in the original work [CCDGB, CCDGa].

We assume that the input data is a time series of scalar fields. Each scalar field is given as a 2D grid-sample of scalar values with the sampling positions being identical for every time step. As we did in Chapter 3, we tessellate the 3D space, which consists of the 2D space and time in this chapter. Delaunay triangulation is employed for tessellating the 3D space into a set of tetrahedra.

Note that the JCN construction [CD14] assumes linear interpolation of data values within each tetrahedron.

### 5.5.2 Constructing the Joint Contour Net

We construct the JCN with the rasterization algorithm [CD14], as in Chapter 3. Here, a strategy is necessary that locates the local extrema in the domain. We achieve this by obtaining the inverse mapping, which maps each JCN node (lying in the range) to the corresponding sampling points (in the domain).

To do so, we start the JCN construction with finding the destination of the sampling points in the framebuffer. Here, we need to guarantee that we find the sampling point  $p$  in the quantized data value  $f_r(p) \in \mathbf{R}^2$  as we have obtained above, which may not be the case if we only rasterize the triangles. Since we also rasterize all the points in the triangles as described in Chapter 3, this holds: i.e.  $p$  is always mapped to  $f_r(p)$  after the rasterization.

### 5.5.3 Simplifying the Jacobi Set through Reeb Skeleton Simplification

In order to identify the Jacobi set segments that are to be simplified, we first calculate the persistency of each Reeb skeleton node, and map it to each Jacobi set segment. Each Reeb skeleton node has a set of JCN nodes, each of which becomes active if a Reeb skeleton node containing the JCN node becomes active. A Jacobi set segment becomes active if one of its containing tetrahedra are active, i.e. a JCN node containing the tetrahedron is active. For each regular Reeb skeleton node, we define its persistency as the persistency of the sum of the birth node. As discussed in Section 5.4.3, we choose the time-integral volume as the persistency. This is achieved in the following manner. For each local extremum in a time step, we find the closest enclosing contour containing the saddle, as in Figure 5.1. The idea

is to give the persistency as the volume swept by the enclosing contour. We approximate the volume with *active cell count* [CDD06], i.e. the number of tetrahedra touching the volume. The active tetrahedra are obtained for each JCN node. To do so, we record the tetrahedra when they are projected in the JCN construction algorithm, and create a mapping from each JCN node to the tetrahedra.

#### 5.5.4 Extracting the Jacobi Set

Although an algorithm for extracting the Jacobi set [EH04] exists, it does not take into the account the quantization of the values. Therefore, it is not applicable to our approach. Indeed, for tetrahedra that are degenerate, i.e. more than two of its vertices are mapped to the same function value  $(f_1, f_2)$ , locally checking for the deviations of values cannot determine the singular fiber type such as birth.

We therefore find the local extrema by checking the graph structure of the JCN. Recall that each of the JCN node corresponds to a fiber component. Here, since  $f_2(x, y, t)$  is the time, each fiber component is a contour of some scalar value  $f_1(x, y, t)$ , in a single time step  $t$ . Therefore, we find a local extremum by finding a birth fiber. We model the image of a local extremum, as a JCN node  $i$  that has no adjacent node of function value  $(f_1(i) + 1, f_2(i))$ , where  $f_j(i)$  denotes the function value of the JCN node  $i$ .

Edelsbrunner et al. [EH04] showed that the local extrema move only along the edges of tetrahedra. Although Edelsbrunner et al. did not quantize the function values but instead perturbed them, the location of local extrema can approximate our quantized version. Therefore, we approximate the local extrema by the edges of tetrahedra: for each local extremum node  $i$  of the JCN, we find the set of local extrema (in our sense) as a set of tetrahedron edges  $e$ :  $f^{-1}(i) = \{e \mid e \text{ is incident to a tetrahedron point which is mapped to } i\}$ . A remaining problem is that the set  $f^{-1}(i)$  can have multiple edges as its

elements. Our experiments in Section 5.6 indeed suffered from this problem, but yet the visualization results were practical.

## 5.6 Results and Evaluation

We tested our approach by simplifying the local extrema with a few plasma fluid datasets and an analytic function. The datasets were obtained by courtesy of Dr. Shinya Maeyama from the Japan Atomic Energy Agency.

Firstly, we evaluated the effectiveness of our approach with a simple fluid dataset where two local extrema merge to one. Secondly, we analyzed a larger dataset with vortices of varying sizes. Finally, we evaluated the effect of quantization, with an analytic function as the ground truth. There, we provide two measures such as the number of local extrema and their continuity for quantitative analysis.

### 5.6.1 A Simple Fluid Dataset

Figure 5.3 shows a simple dataset: two local extrema merging to one. Here, we learned that the local extrema in the original dataset were captured in the JCN, and our persistency indeed corresponds to the scale in this example. In Figure 5.3, we varied the persistency threshold from 0 to 1. In Figure 5.3 (d) and Figure 5.3 (e), we increased the persistency to 0.05, and observed that the two major local extrema became omitted. This accounted to the fact that the two relatively small vortices merged to a large vortex in the dataset. We therefore concluded that varying our persistency indeed corresponded to varying the scale of the physical phenomena.

However, the local extrema during the merging event of the two local extrema were lost in the results. For an explanation, we found that the fibers became significantly complex during the merging events in the fluid simulation dataset, and separated themselves to many small fiber compo-

nents. Even worse, merging events tended to have a rapid change in the scalar values. Indeed, such a merging of local extrema corresponds to a cusp, which contains a cubic term in the canonical equation, as shown in [STS<sup>+</sup>14]. Therefore, the Reeb space had many small 2D sheets that are connected with each other. Then, the quantized approximation with the JCN should have often lost such small details. Consequently, the local extrema were not captured.

### 5.6.2 A Fluid Dataset with Abundant Vortices

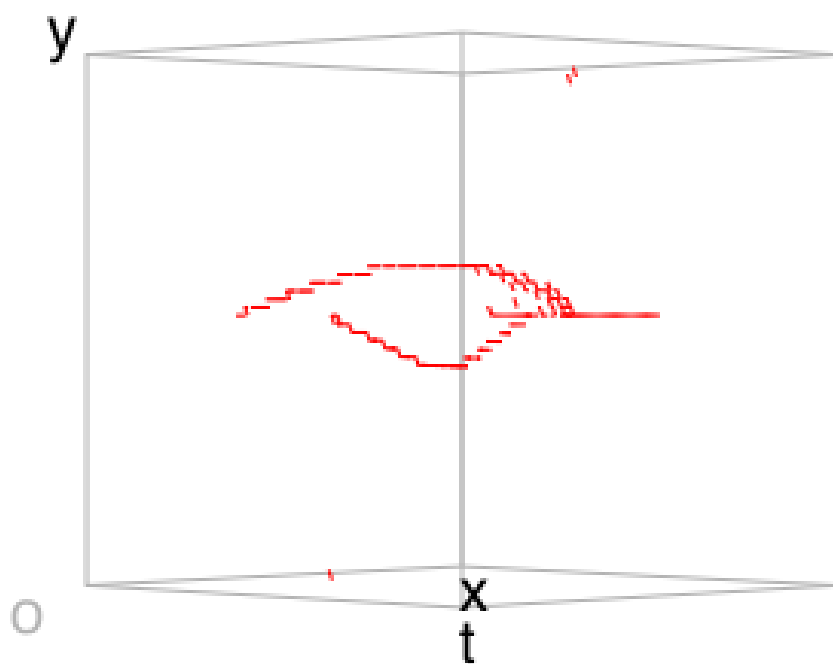
Figure 5.4 shows the visualization results of a plasma turbulent fluid with more vortices. Here, the dataset had  $64 \times 64$  spatial resolution for 20 time steps, and the data values were quantized to 128 values in the JCN. As we raised the threshold, local extrema with small persistency disappeared from the simplification as seen in Figure 5.4 (a–f). Some local extrema in the original dataset Figure 5.4 (g–h) were not captured. These corresponded to the limitation of the approximation of the Reeb space with its quantized JCN. These results indicate that the user would indeed be able to select the local extrema of interest by controlling the threshold, which corresponds to the size of local extrema, or the scale of vortices in fluids.

### 5.6.3 An Analytic Function

Finally, we show the effect of the choice of the quantization level. Figure 5.5 shows how a choice of the quantization level affects the results. As a ground truth, we chose an analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ , and investigated how well our system extracts the local extrema in this dataset. In the figure, (a) and (b) shows the actual configuration of local extrema in the domain and the range, respectively.

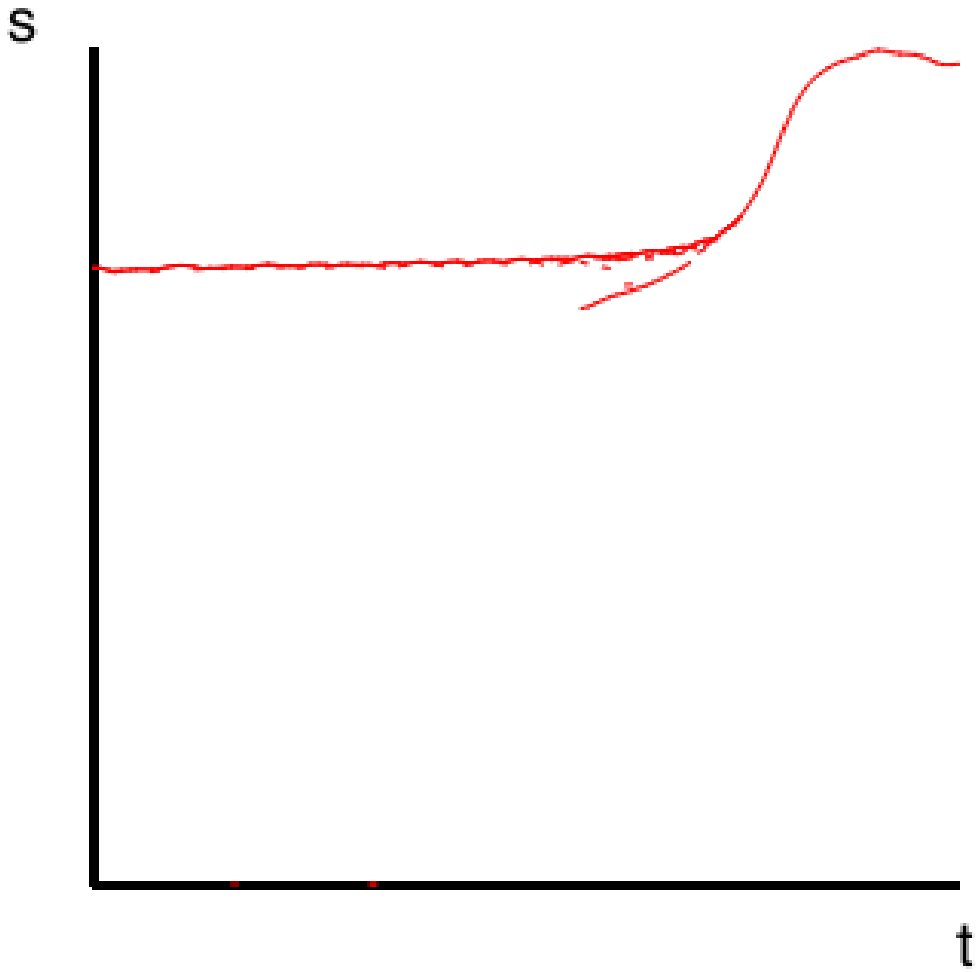
Here, in order to visualize such an effect, we have chosen two indices that provide an overview for understanding the effect of the different resolutions





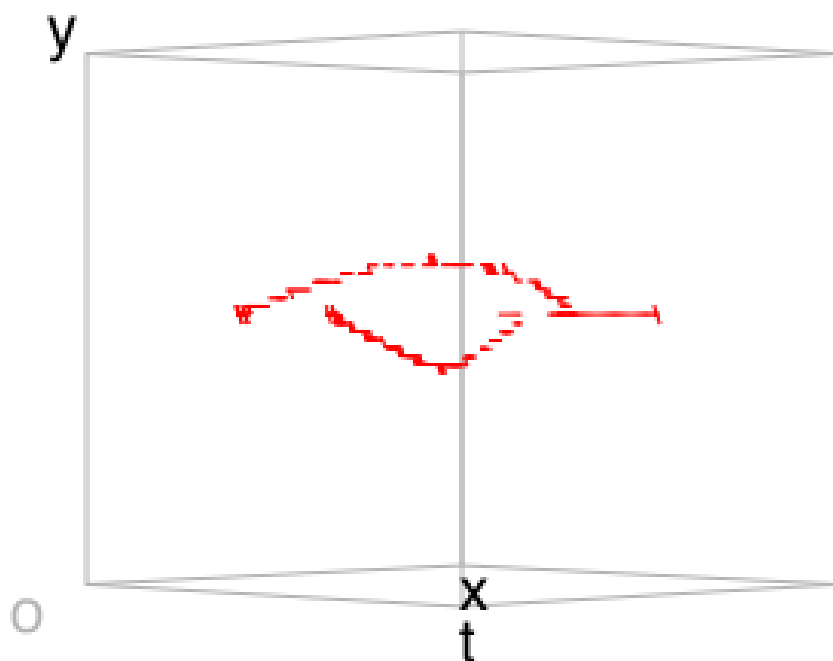
(a)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (a) The original local extrema in the domain.



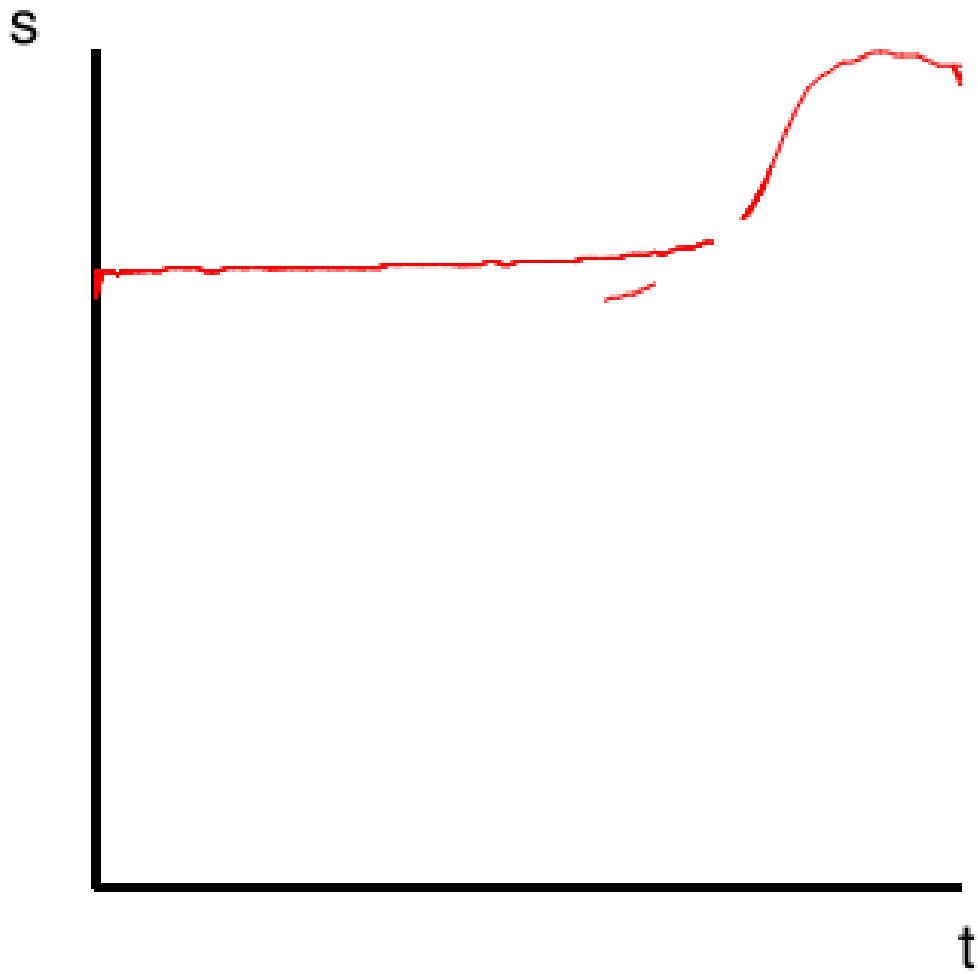
(b)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (b) The original local extrema, plotted to the range.



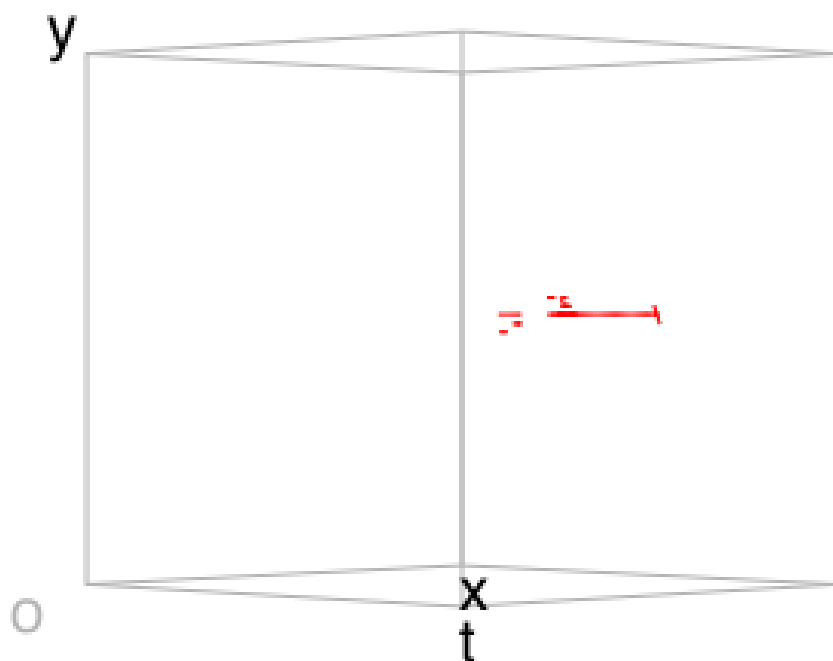
(c)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (c) The extracted local extrema, in the domain.



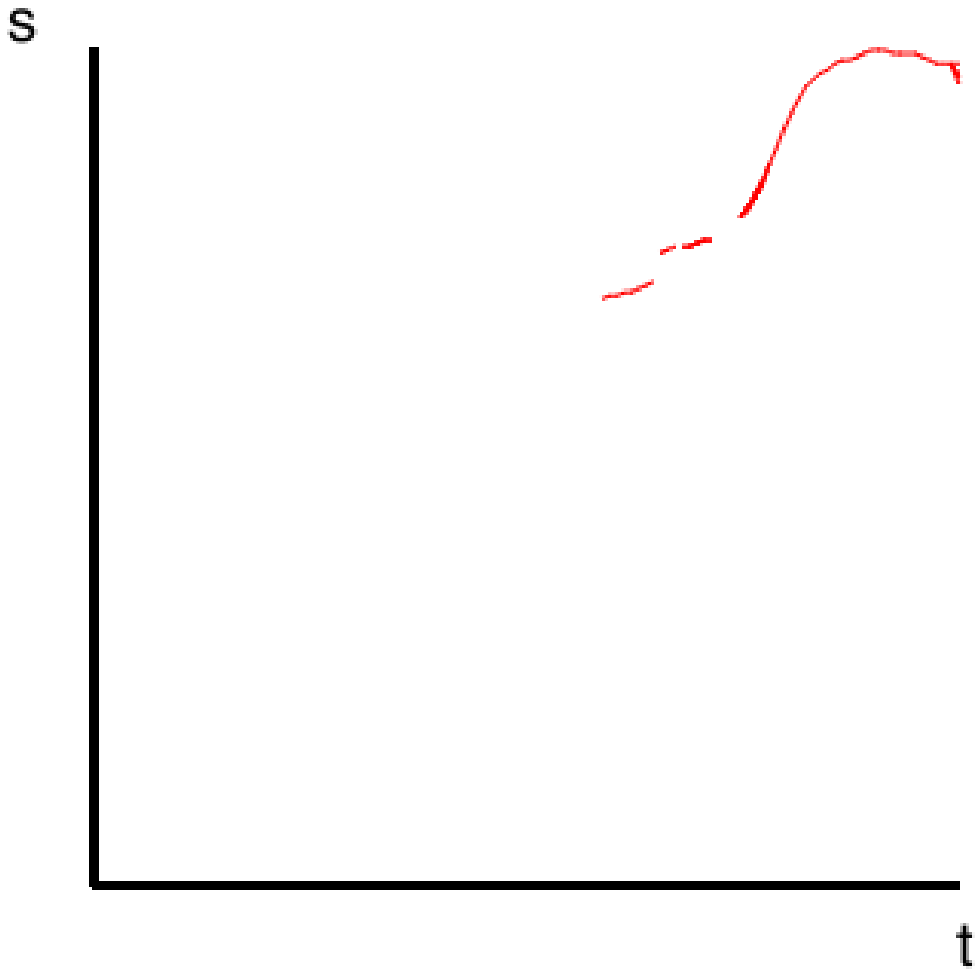
(d)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (d) The extracted local extrema, plotted to the range.



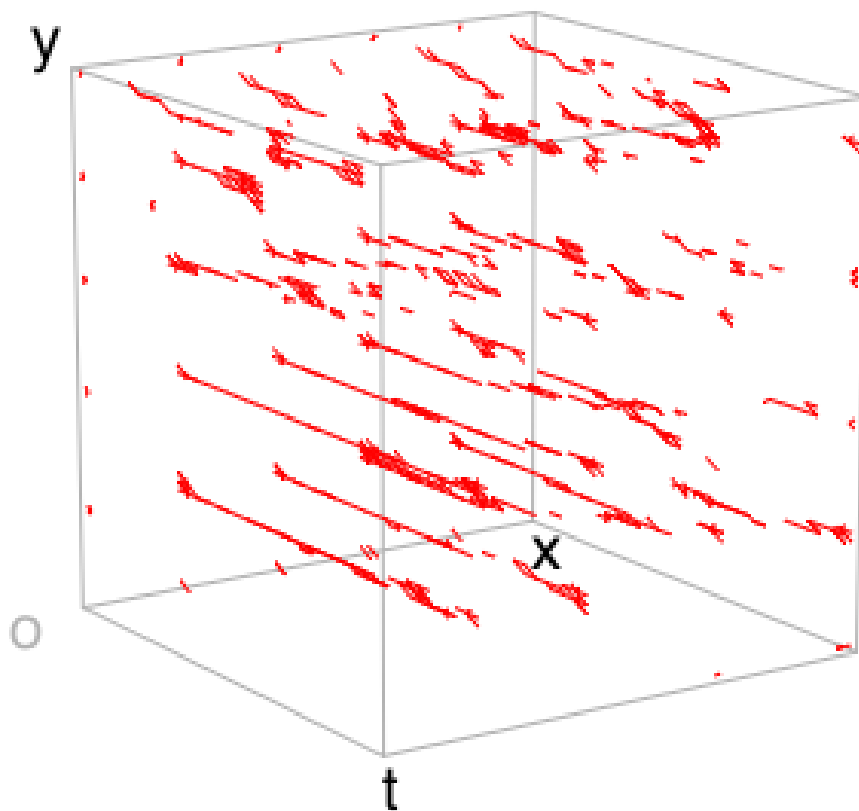
(e)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (e) The extracted local extrema with persistency value 0.05, in the domain.



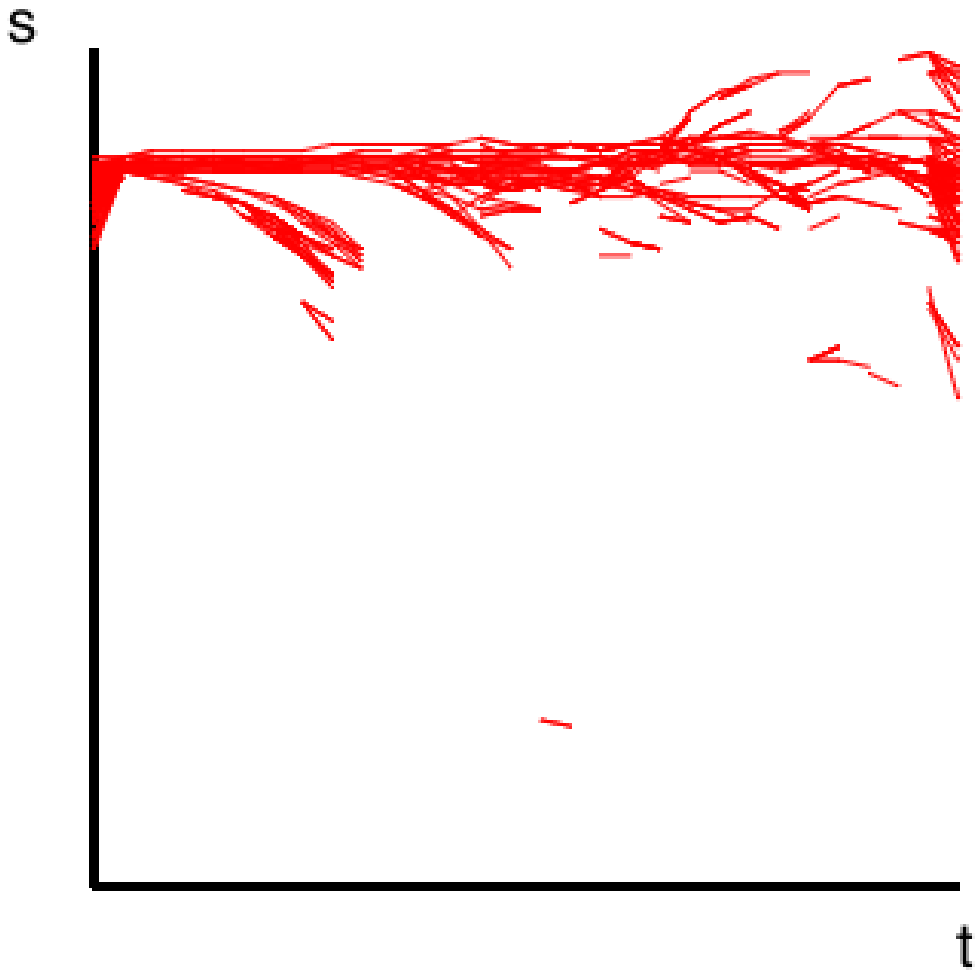
(f)

Figure 5.3: A simple fluid dataset. Two local extrema merged to one. Note that the two local extrema had an identical scalar value. (f) The extracted local extrema with persistency value 0.05, plotted to the range.



(a)

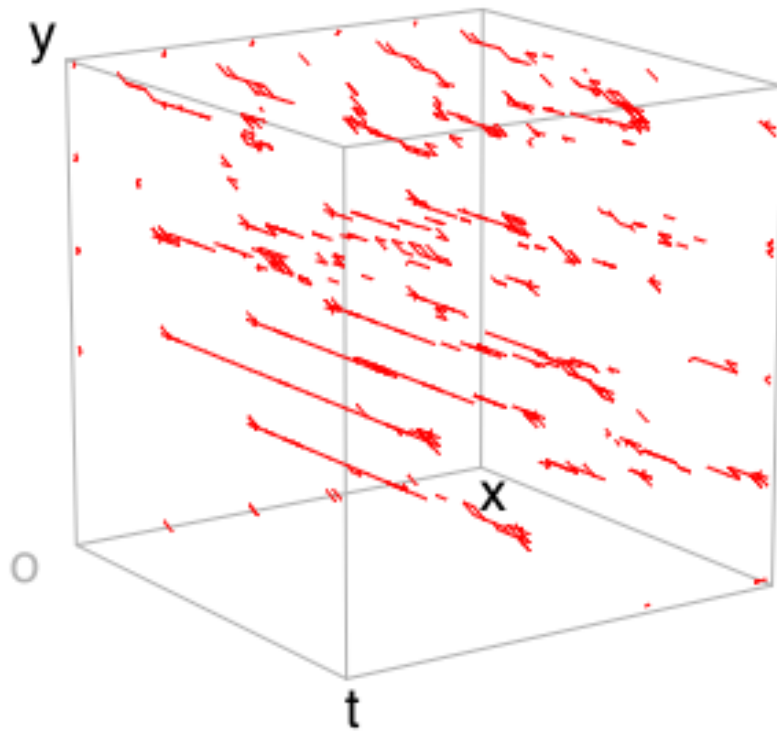
Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (a) The domain. The persistency threshold was 0.



(b)

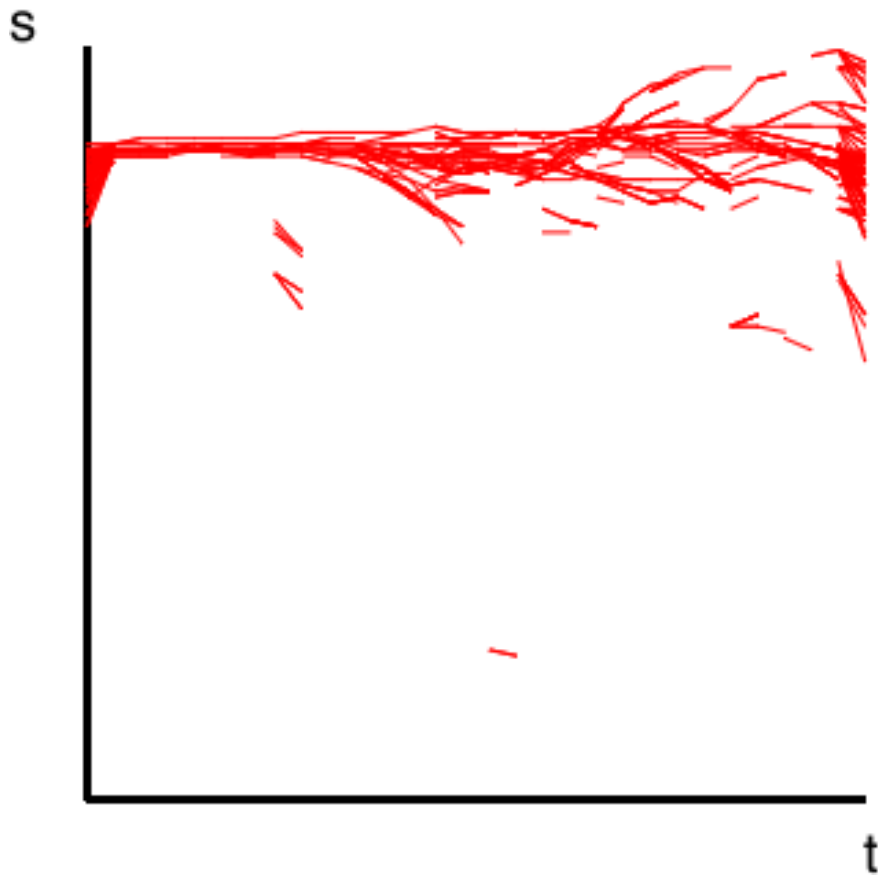
Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (b) The range. The persistency threshold was 0.





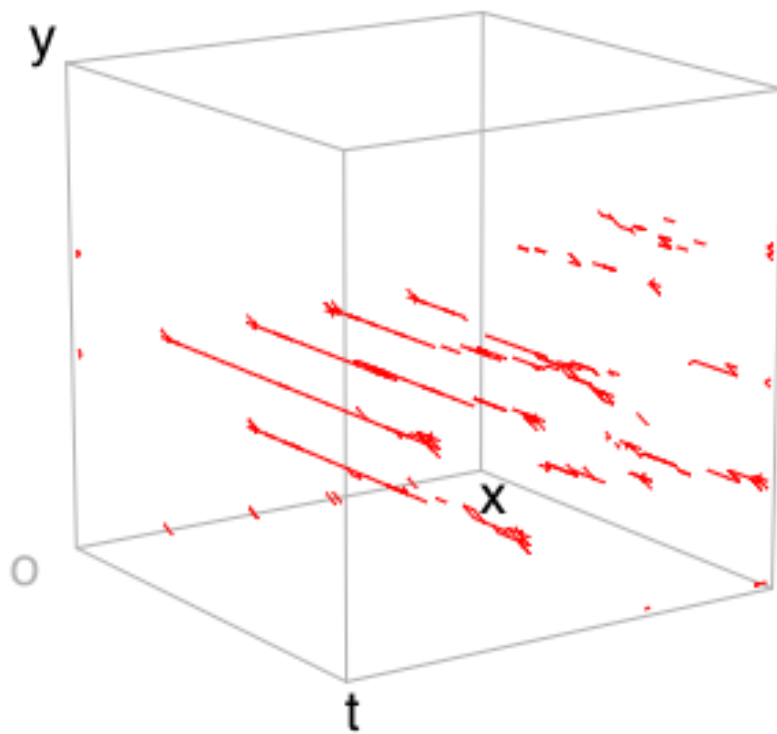
(c)

Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (c) The domain. The persistency threshold was 0.2.



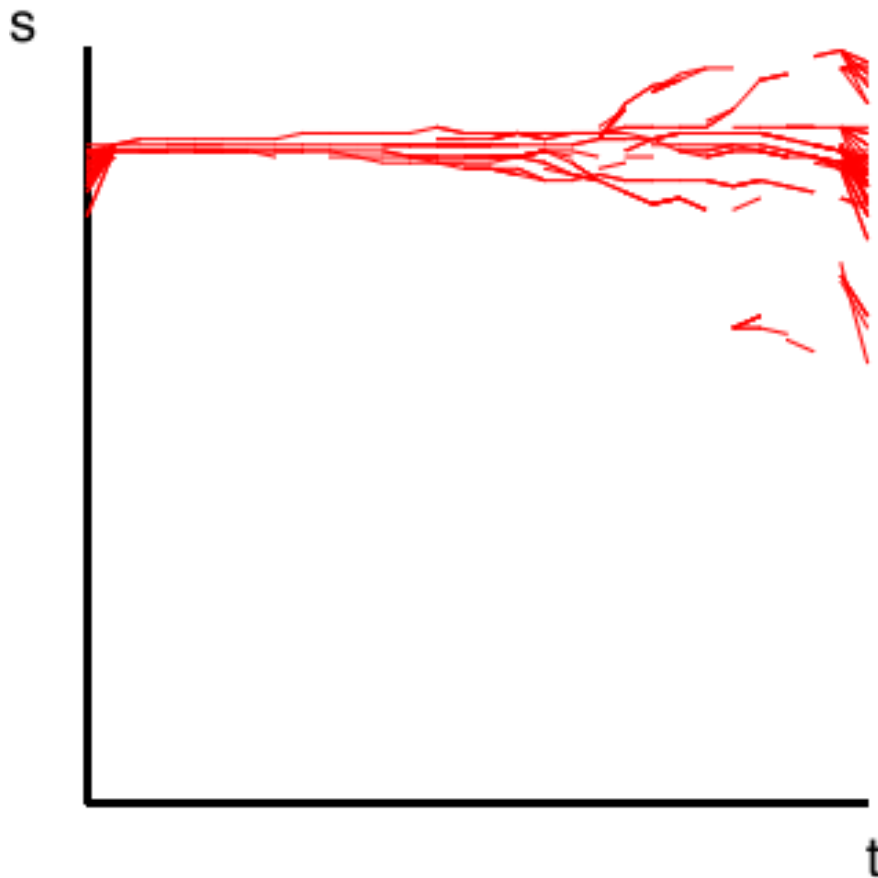
(d)

Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (d) The range. The persistency threshold was 0.2.



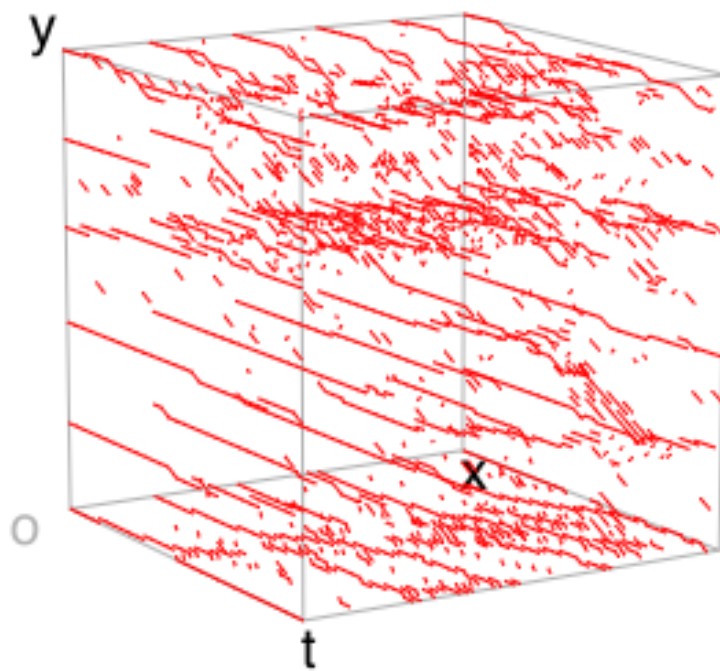
(e)

Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (e) The domain. The persistency threshold was 0.85.



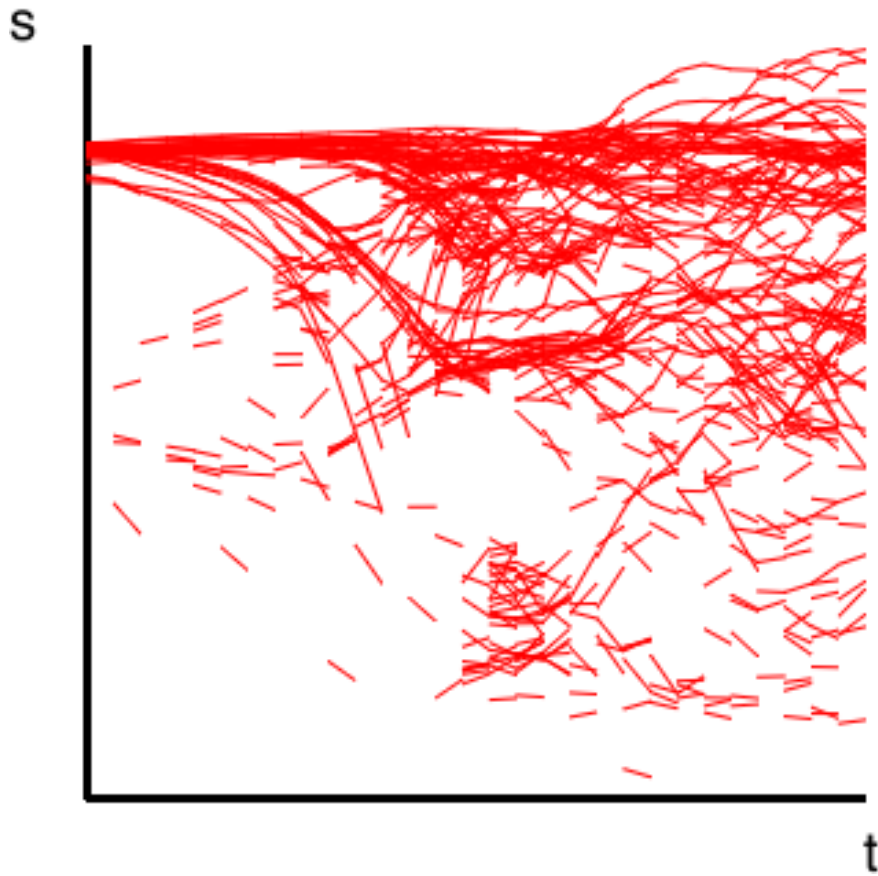
(f)

Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (f) The range. The persistency threshold was 0.85.



(g)

Figure 5.4: The original Jacobi set (a–b), and the simplified versions (c–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (g) The original Jacobi set in the domain.



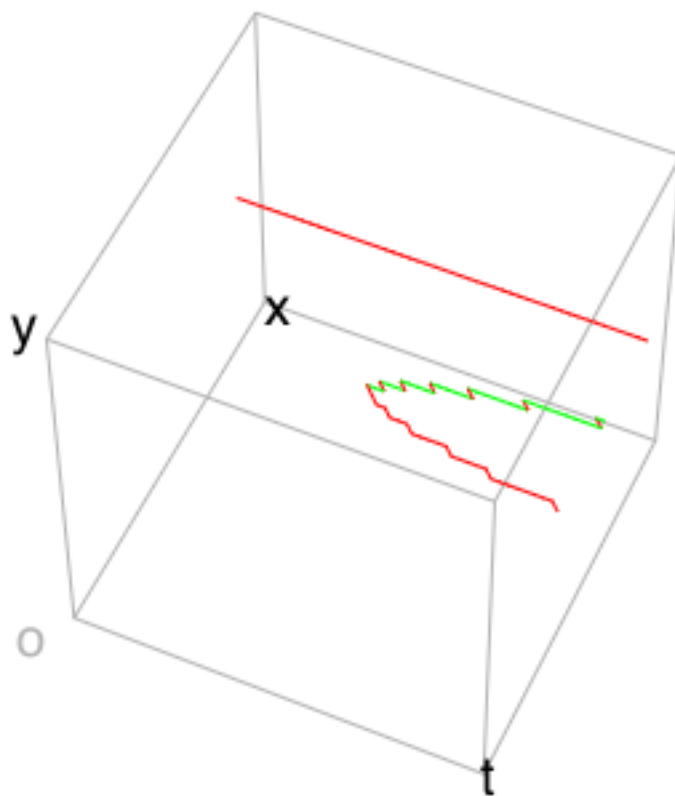
(h)

Figure 5.4: The simplified Jacobi set (a–f) and the original Jacobi set (g–h).  $s$  is the scalar value  $f_1$ , and  $t$  is the time. (h) The original Jacobi set, plotted to the range

of the quantization. The first index is the *number of local extrema*. In general, a small number of local extrema corresponds to less duplication. In fact, our approach tends to capture redundant local extrema, since it identifies the local extrema as the inverse image of birth nodes in the JCN. The second index is *the connectivity ratio*. This index is calculated as the ratio of an local extrema edge  $e_{t,t+1}$  in tetrahedron to connect to another local extrema edge  $e_{t+1,t+2}$  in the next time step. Here, note that we omit checking the previous edge  $e_{t-1,t}$ , since otherwise we double-count the connection. We also integrated this view into the interface for providing the user with an opportunity to find the quantization levels that are appropriate for the investigation of a user specified dataset.

The results are shown in Figure 5.6, where a color glyph with dark red indicates low value, and that with light red indicates the opposite. Here, we can see that the number of local extrema decreased when the quantization level of the scalar value increased. We also found that the difference in the quantization of time were not significant for the chosen analytic function. Similarly, the connectivity ratio decreased by raising the resolution of the scalar value,  $f_1$ . This happened as the precision of the captured local extrema increased. Such a precise extraction necessarily reduces the volume of the inverse image of local extrema, as can be seen in Figure 5.5. The smaller volume of inverse image causes the captured local extrema, which is actually tetrahedra edges, to become fewer. Here, since most of the edges are connected inside the inverse image, the connectivity ratio tended to be higher in larger inverse images, or smaller quantization levels.

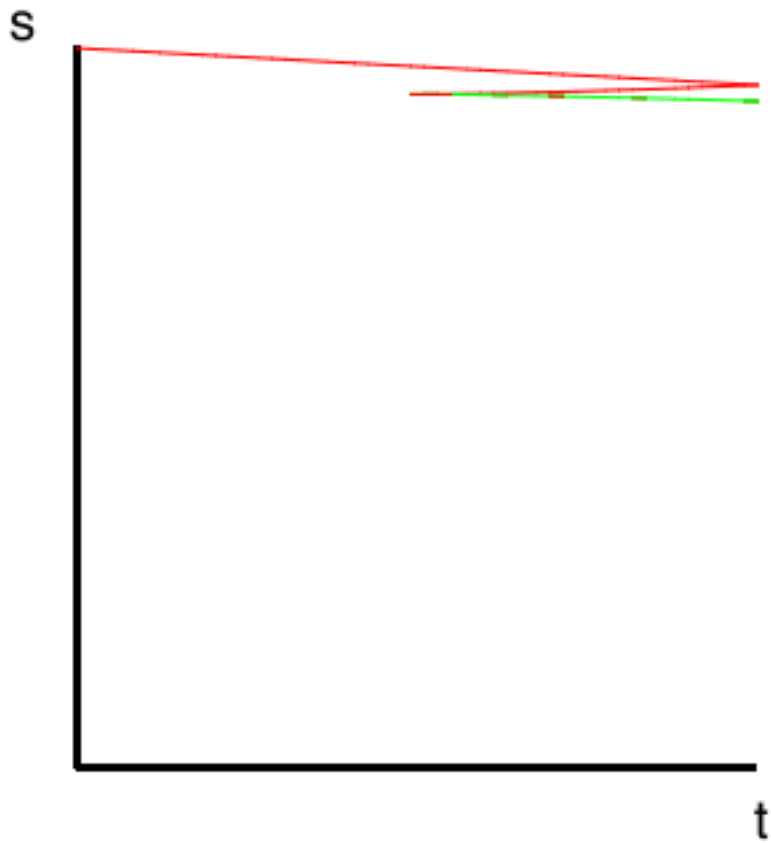
Further evaluation is found in Chapter 6, as an evaluation as a whole for the proposed fiber topology extraction approach.



(a)

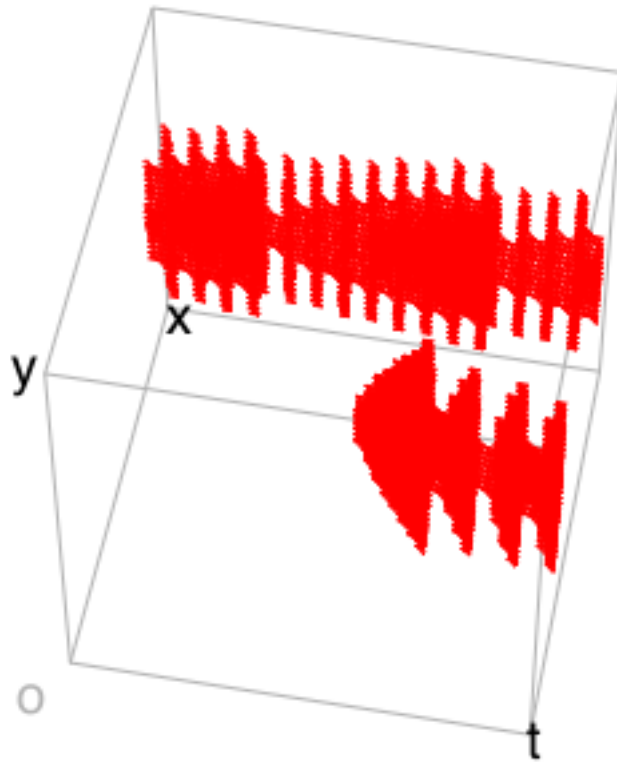
Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ . (a) The original local extrema in red, in the domain. For reference, saddles were shown in green.





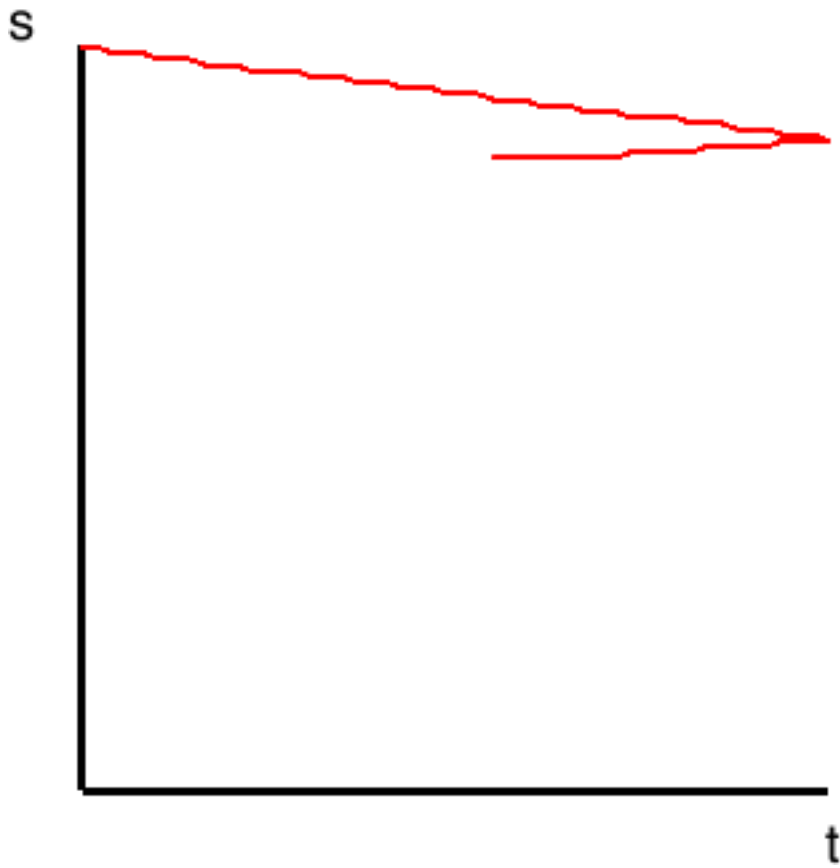
(b)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
 (b) The original local extrema in red, in the range. For reference, saddles were shown in green.



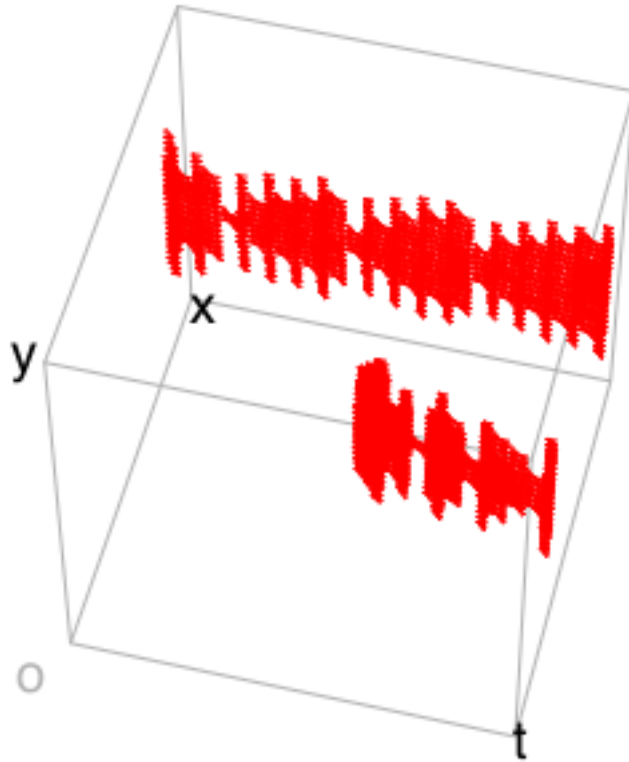
(c)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
(c) The extracted local extrema in the domain, with the scalar resolution of 128 levels, and the time resolution of one level per time step.



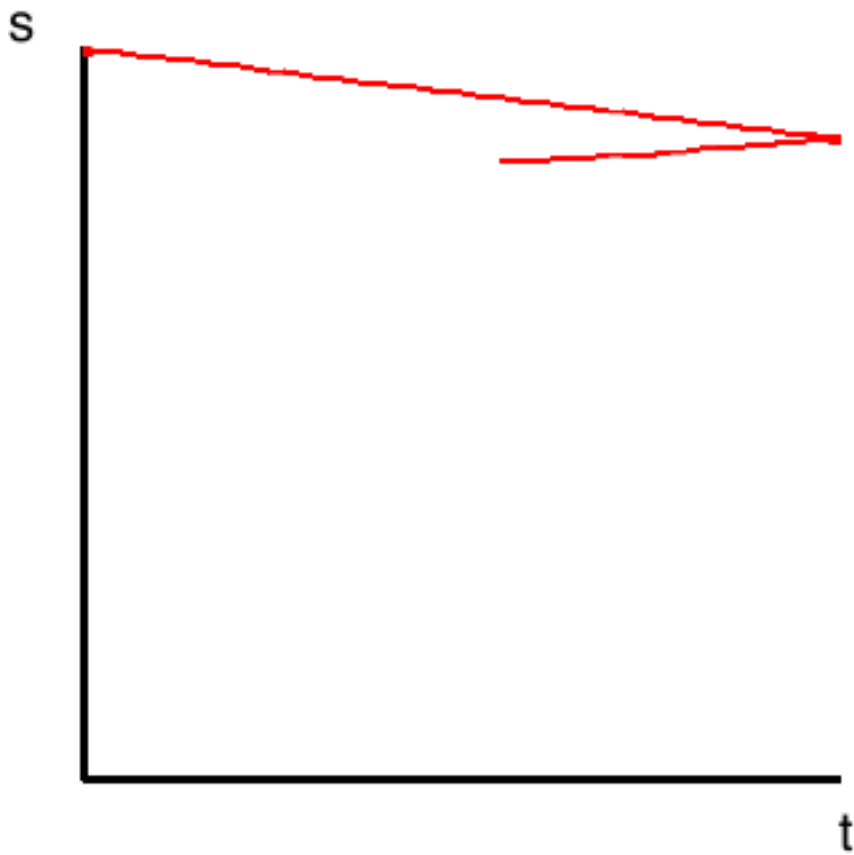
(d)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
 (d) The extracted local extrema in the range, with the scalar resolution of 128 levels, and the time resolution of one level per time step.



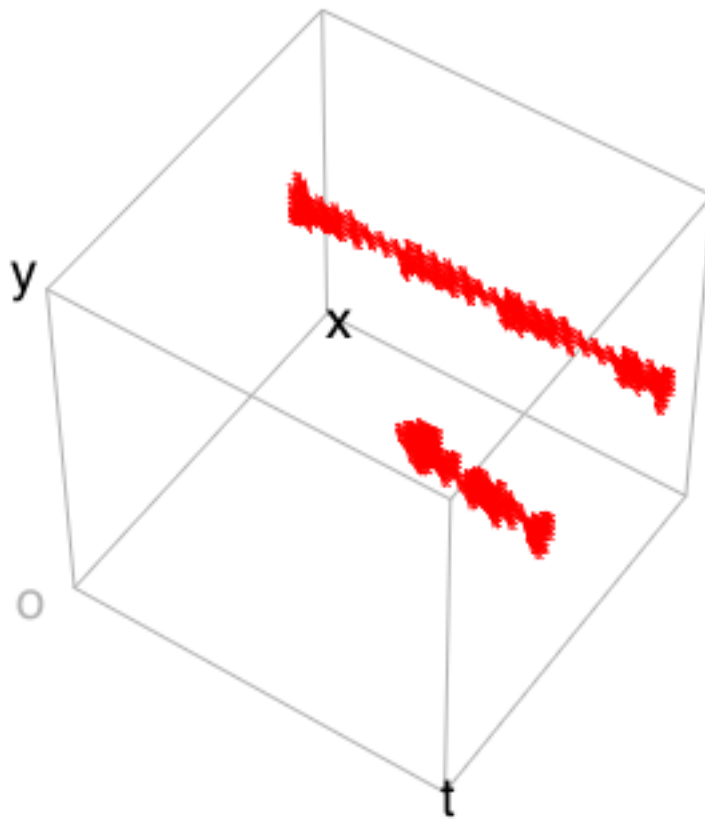
(e)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
(e) The extracted local extrema in the domain, with the scalar resolution of 256 levels, and the time resolution of one level per time step.



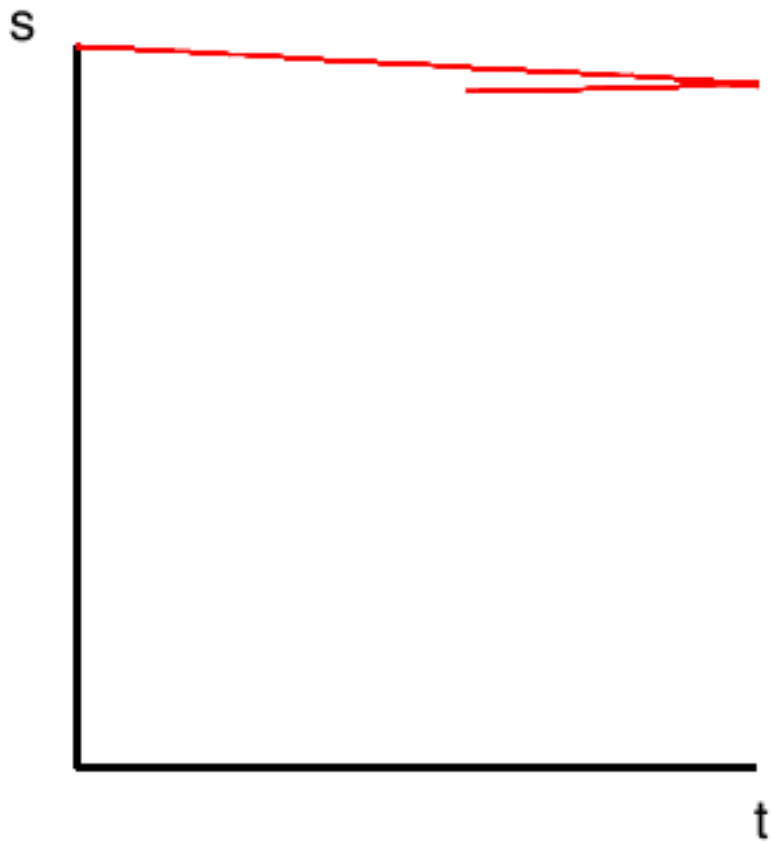
(f)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
 (f) The extracted local extrema in the range, with the scalar resolution of 256 levels, and the time resolution of one level per time step.



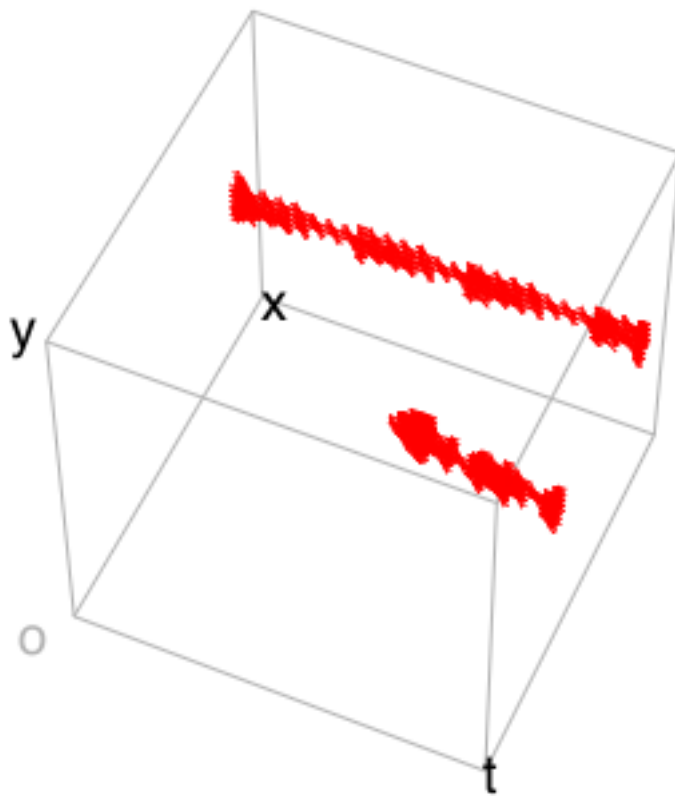
(g)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
 (g) The extracted local extrema in the domain, with the scalar resolution of 512 levels, and the time resolution of one level per time step.



(h)

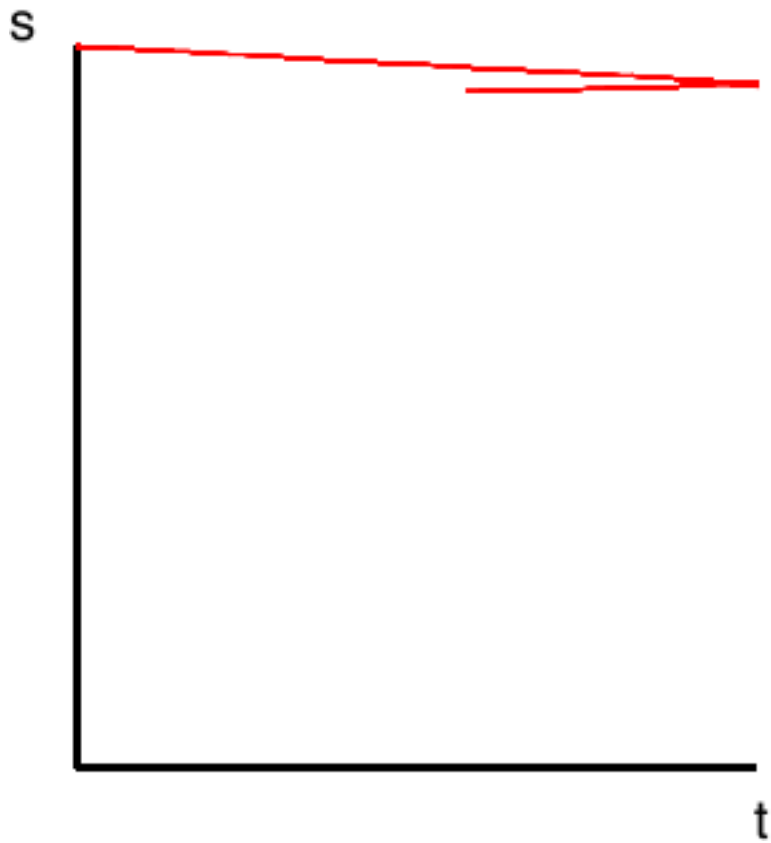
Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
(h) The extracted local extrema in the range, with the scalar resolution of 512 levels, and the time resolution of one level per time step.



(i)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
(i) The extracted local extrema in the domain, with the scalar resolution of 512 levels, and the time resolution of three levels per time step.





(j)

Figure 5.5: An analytic function  $f_1(x, y, t) = (2x^3 + t(-3x + x^2) - x^4 - y^2)$ .  
 (j) The extracted local extrema in the range, with the scalar resolution of 512 levels, and the time resolution of three levels per time step.

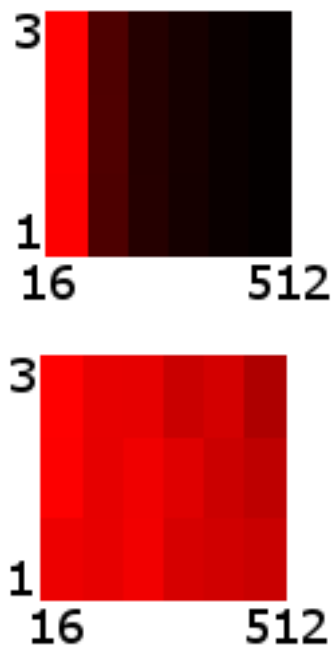


Figure 5.6: The effect of quantization, visualized as color glyphs. Here, the image on the top shows the number of the extracted local extrema. In contrast, the image on the bottom shows the connectivity of the local extrema. For both of the indices, dark red indicates low value, and light red the opposite. The horizontal axis represents the quantizations of the scalar value, which are 16, 32, 64, 128, 256, or 512. On the other hand, the vertical axis represents the resolutions of time per a time step, which are 1, 2, and 3.

# Chapter 6

## Discussion

In this chapter, we discuss the effectiveness of our approach to extracting singular fibers and to visualizing them. Especially, we provide a few analyses on the effect of quantization to the feature extraction, as Section 6.1. Then in Section 6.2, we classify errors that may exist in extracted singular fibers. Furthermore, miscellaneous evaluations are given in Section 6.3.

### 6.1 Resolution of the Quantization

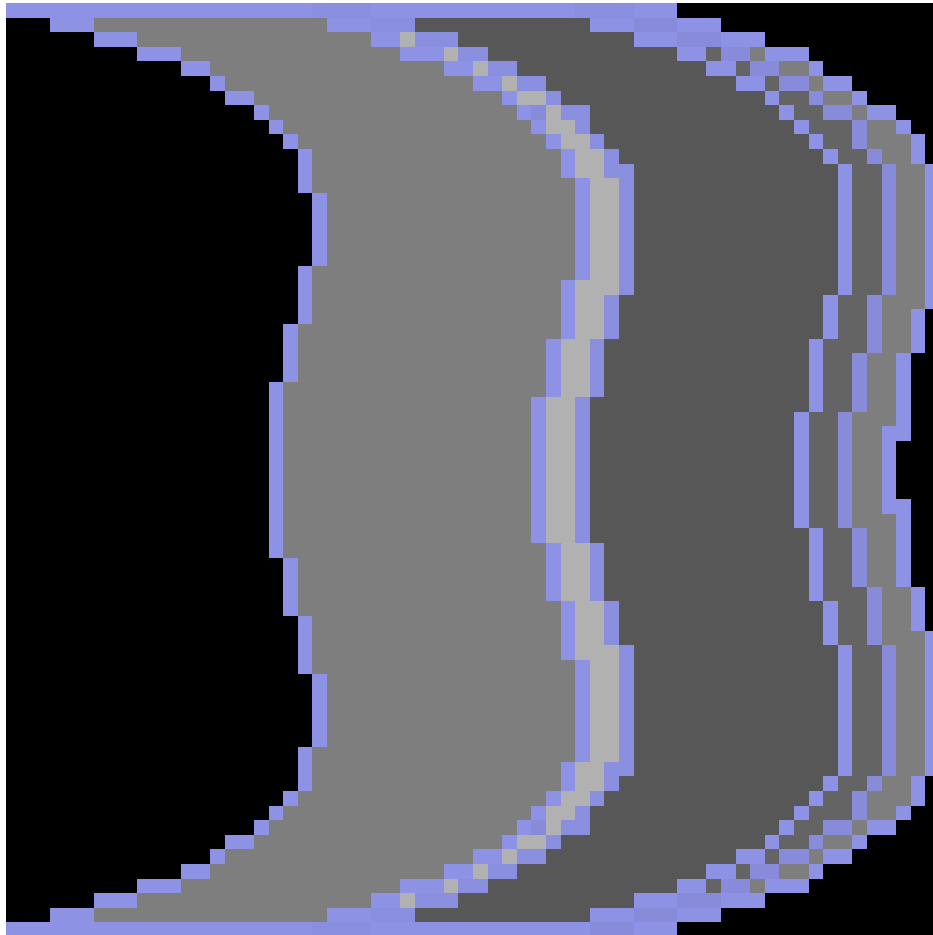
Here, we discuss how the resolution of the JCN impacts our approach. Figure 6.3 and Figure 6.5 shows the visualization results with different resolutions of the range. Recall that, in Figure 5.3 of Chapter 5, the merging of birth fibers (or local extrema) were not captured. The reason for this was the quantization by the JCN. More precisely, the topology of fibers were not represented in the JCN due to a low resolution for describing them. While higher resolutions should succeed in representing the topology in detail, how the resolution actually affect the results is presented below.

### 6.1.1 The Resolution and Identification of the Topological Shapes of Singular Fibers

Figure 6.1 shows the effect of resolution on detecting the topological shapes of singular fibers. Here, the function in Figure 4.1 (the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ ) is used. The color was chosen to show a typical behavior of a miss-identification. Here, the singular fibers were given blue colors except of the miss-identified 8-shaped singular fiber (in red). Surrounding the red pixels are the singular fibers where four fiber components merge to one at once. This reflects the fact that although the equation constricts that four components merge at once, in our numeric system some two components merge together before merging with other two. As this owes to calculation errors, the higher the resolution, the more we find the errors as red pixels. For instance, resolution  $32 \times 32$  (Figure 6.5 (a)) gave no rise to red pixel, while raising the resolutions to  $64 \times 64$  ((b)) and ((c)) increased the number of red pixels, though slightly.

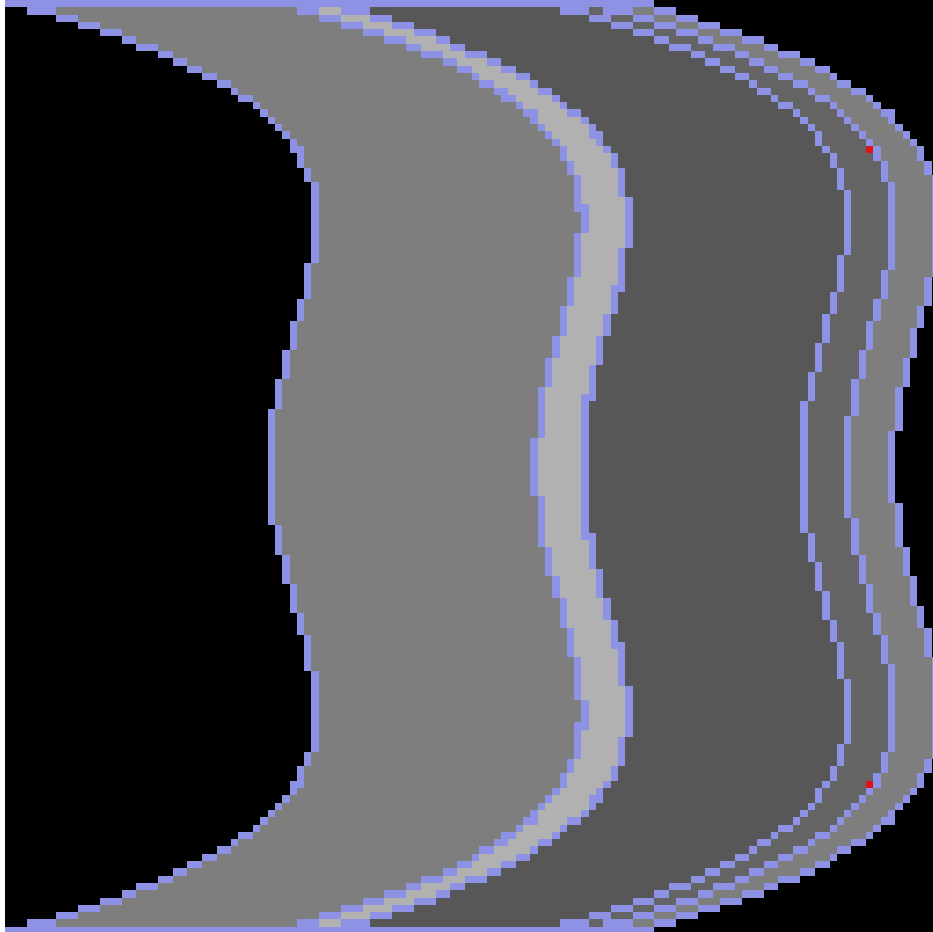
### 6.1.2 The Resolution of Time

We discuss how the resolution along the time axis changes the results. Figure 6.3 shows the results of increasing the JCN's resolution for time. (Figure 6.2 shows the original local extrema in the data set.) We observed that increasing the resolution reduced the duplications of local extrema extracted as the inverse image of the JCN node. Although the continuity of local extrema was extracted better at certain time steps, doing so lost some continuity in other places of time steps. As a conclusion, while increasing the resolution reduces the duplicates of local extrema, the continuity of local extrema was not preserved nor enhanced.



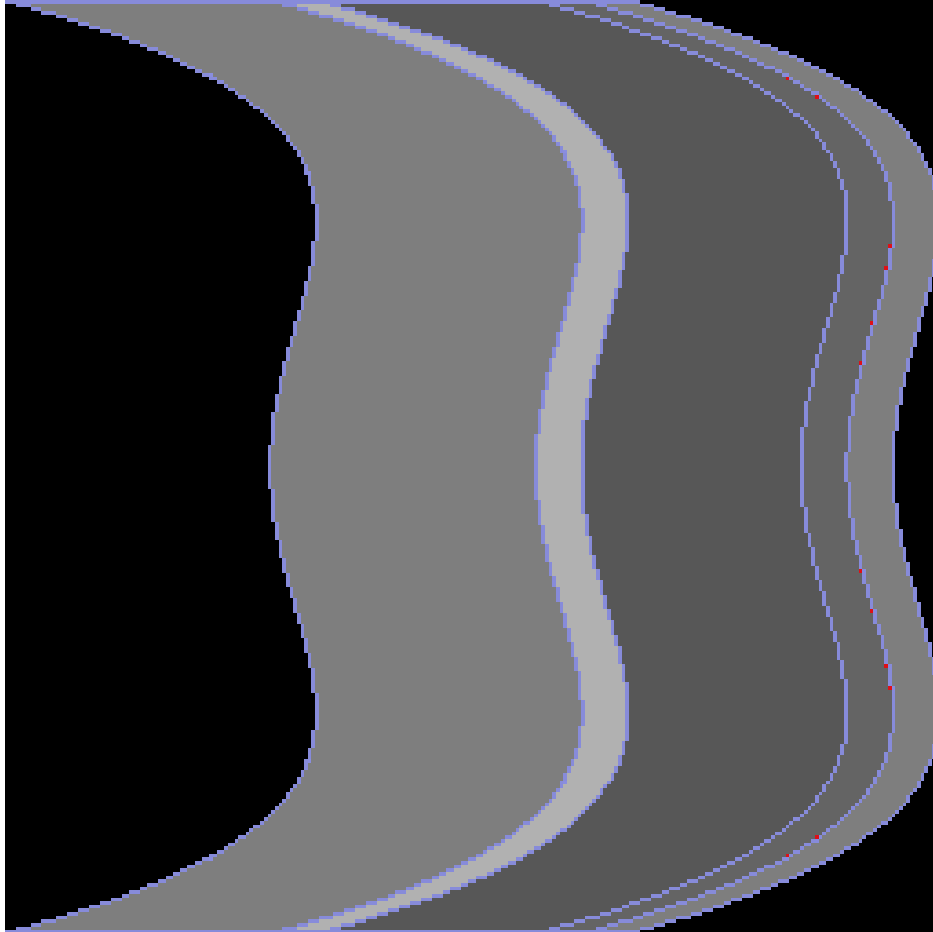
(a)

Figure 6.1: The images of singular fibers identified with different resolutions for the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . Red pixel shows a specific miss-identification, two fiber components merging to one, while four components must merge to one according to the equation. (a) Resolution of  $64 \times 64$ .



(b)

Figure 6.1: The images of singular fibers identified with different resolutions for the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . Red pixel shows a specific miss-identification, two fiber components merging to one, while four components must merge to one according to the equation. (b) Resolution of  $128 \times 128$ .



(c)

Figure 6.1: The images of singular fibers identified with different resolutions for the tangle cube function  $f(x, y, z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ . Red pixel shows a specific miss-identification, two fiber components merging to one, while four components must merge to one according to the equation. (c) Resolution of  $256 \times 256$ .

### 6.1.3 The Resolution of Scalar Field

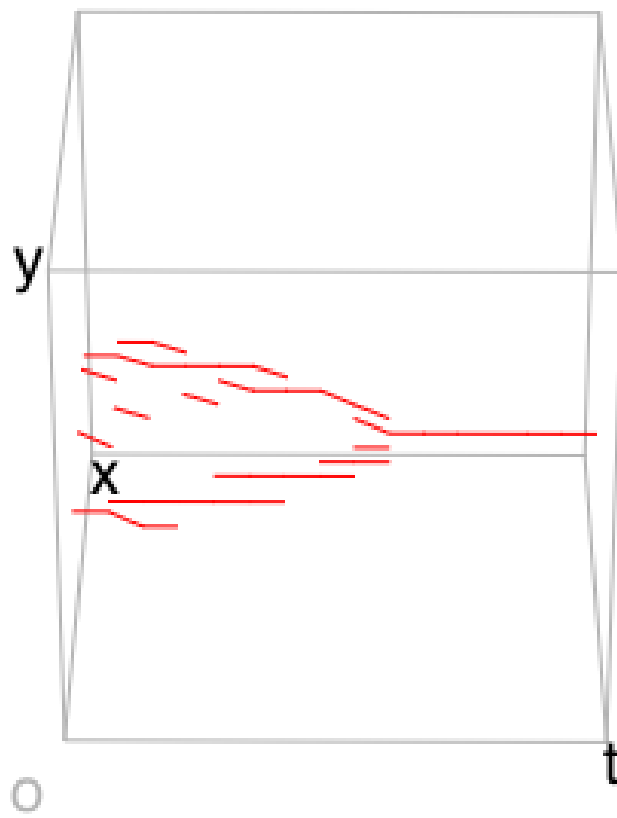
Here, we discuss how the resolution along the scalar value axis changes the results. Figure 6.5 shows the results of changing the JCN's resolution for scalar fields. (Again, Figure 6.2 shows the original local extrema in the data set.) In the visualization results, we observed a trade-off between removing the duplication of local extrema extraction vs. extracting the local extrema: if we raise the resolution to resolve the duplication of local extrema in the domain, the approach fails to capture local extrema. Figure 6.5 demonstrates this trade-off. In Figure 6.5 (a–b), the resolution of the scalar value was low (128 levels of quantization). Here, notice that the merging event of the two local extrema were captured without a gap along the time axis ((a–b)). However, due to the low resolution of the quantization of scalar values, the proposed approach classified excessive tetrahedron edges in the domain as local extrema, as in (b). The rest of Figure 6.5 further supports this trade-off. In (c–d) we duplicated the resolution of quantization to 256, in the hope to get rid of the redundancies of local extrema in (a–b). The finer resolution actually reduced the redundancies, as in (c). However, the time-evolution of local extrema was disconnected in the middle of the time. For verification, we again raised the resolution to 512, and observed the same effect.

## 6.2 Classification for Errors in Extracted Singular Fibers

In this chapter, we classify possible errors in extracted singular fibers. Such errors can occur in each phase of our singular fiber extraction, which were explained in Chapter 3. There, the phases were sampling, tessellation, graph construction, and singular fiber identification. Here, we argue these phases in this order.

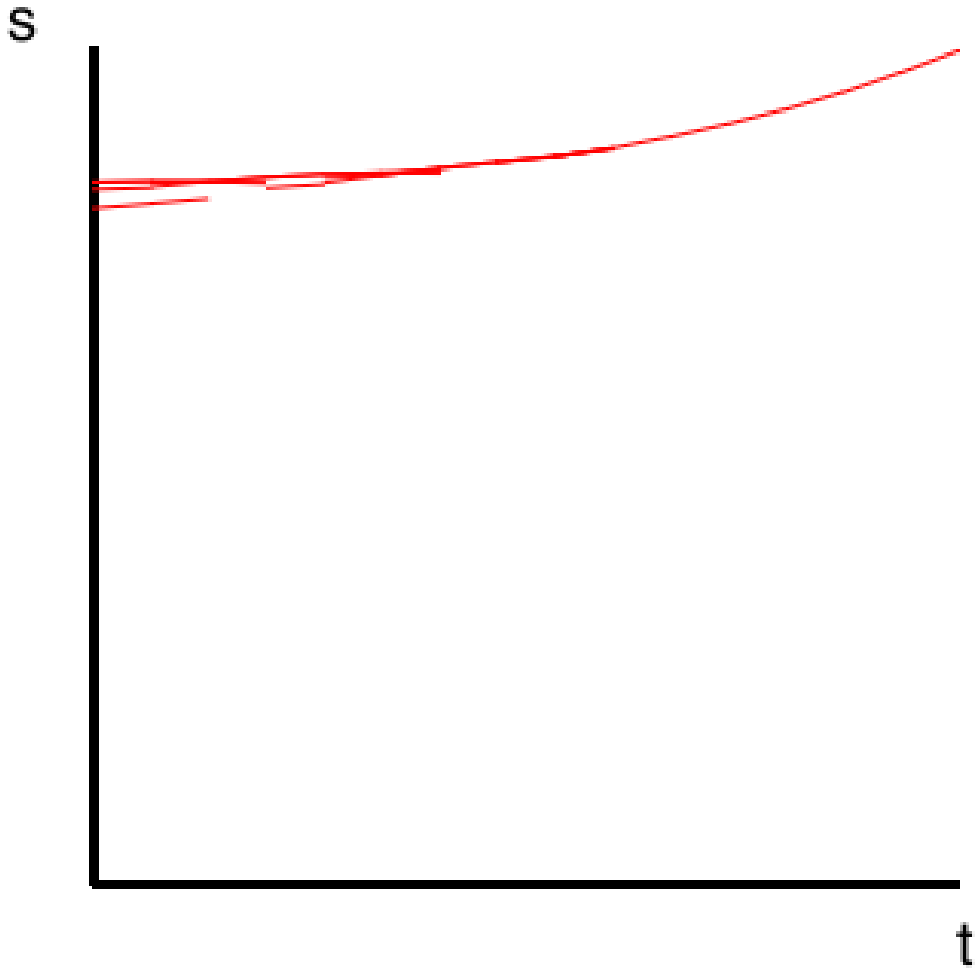
In the sampling phase, we determine the data value of each sampling





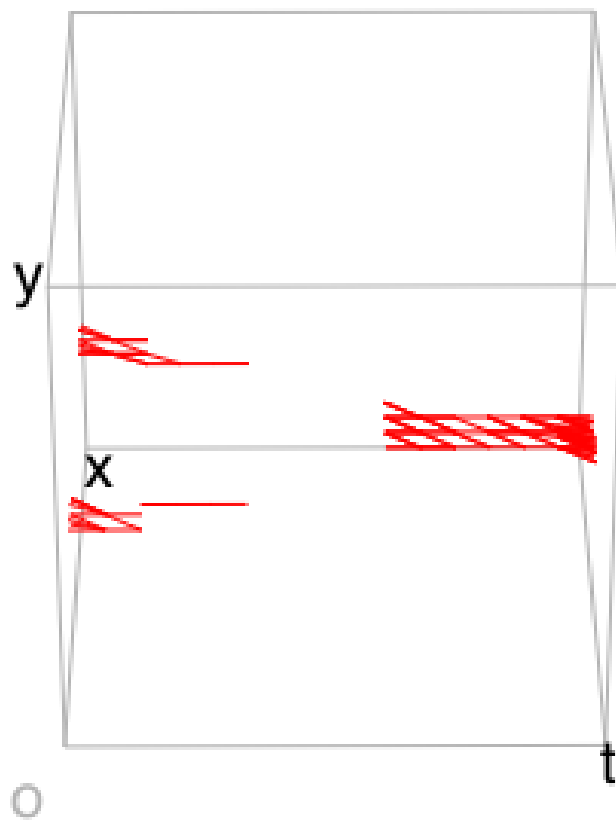
(a)

Figure 6.2: Birth fibers merging. This is a focused view of Figure 5.3. (a) The original birth fibers in the domain.



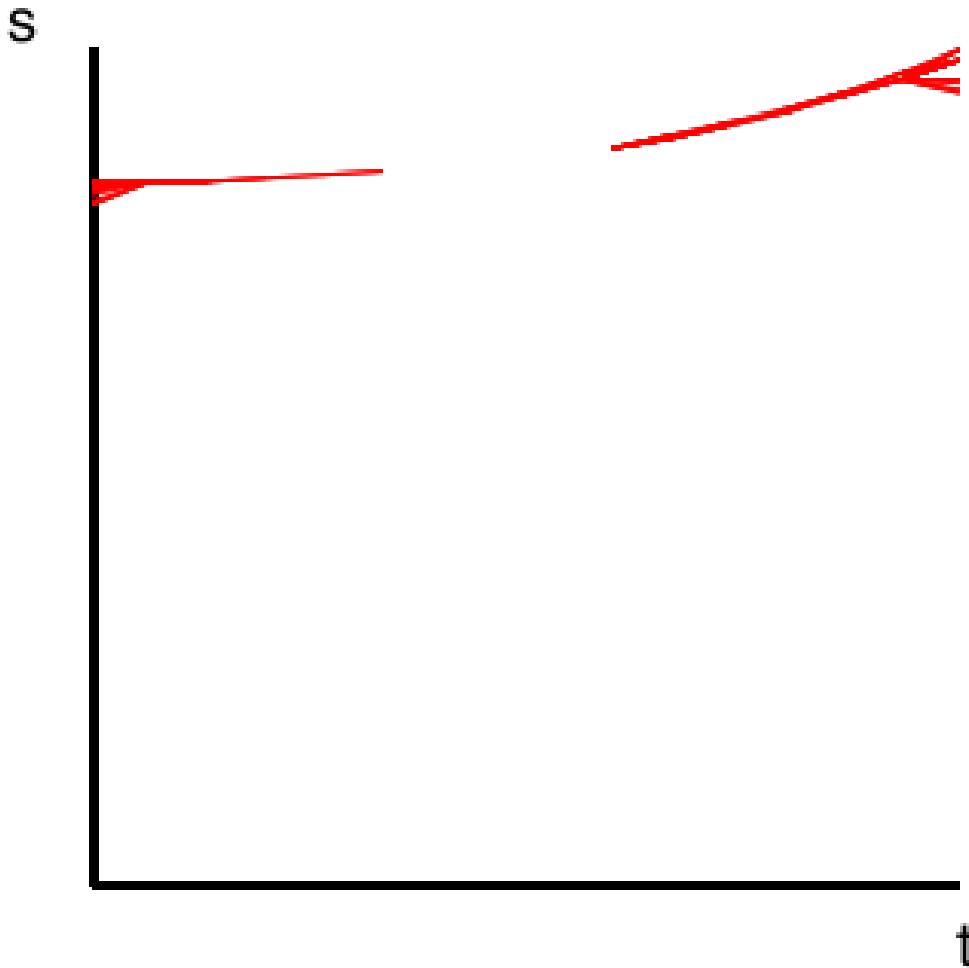
(b)

Figure 6.2: Birth fibers merging. This is a focused view of Figure 5.3. (b) The original birth fibers in the range.



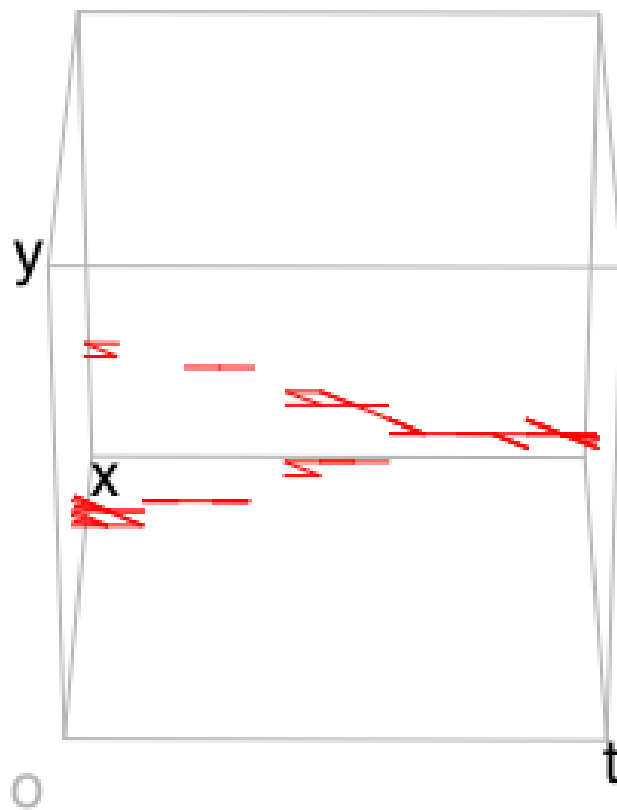
(a)

Figure 6.3: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (a) The birth fibers captured with the JCN, with the scalar value being quantized to 256 levels, and the resolution of time being as large as the number of time steps. The domain.



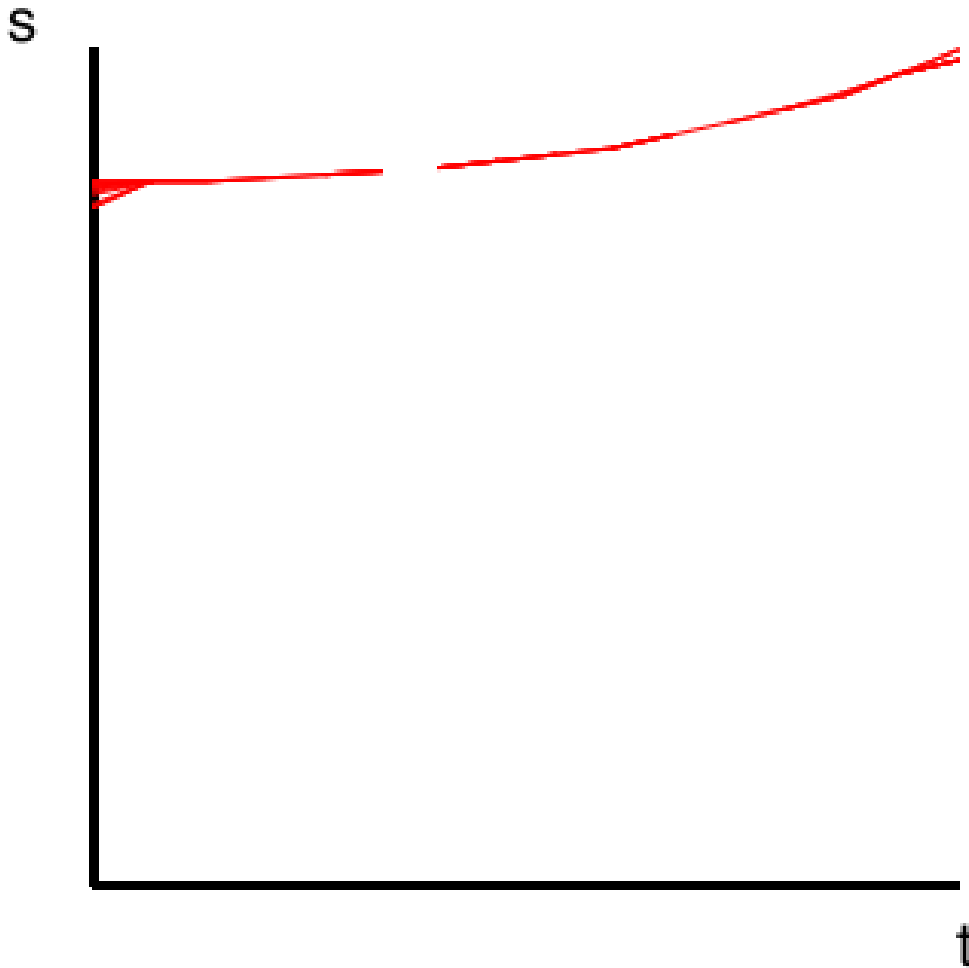
(b)

Figure 6.3: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (b) The birth fibers captured with the JCN, with the the scalar value being quantized to 256 levels, and the resolution of time being as large as the number of time steps. The range.



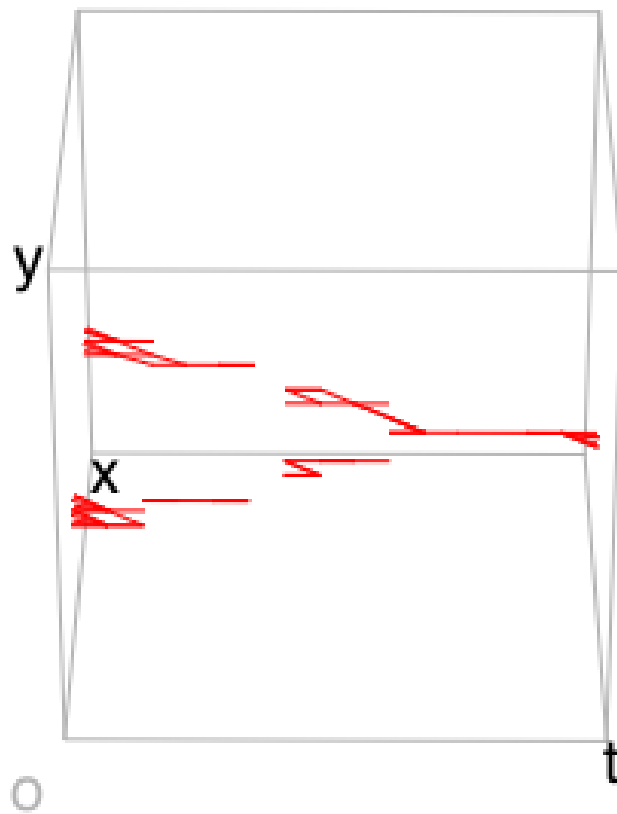
(c)

Figure 6.4: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (c) The birth fibers captured with the JCN, with the scalar value being quantized to 256 levels, and the resolution of time being twice as large as the number of time steps. The domain.



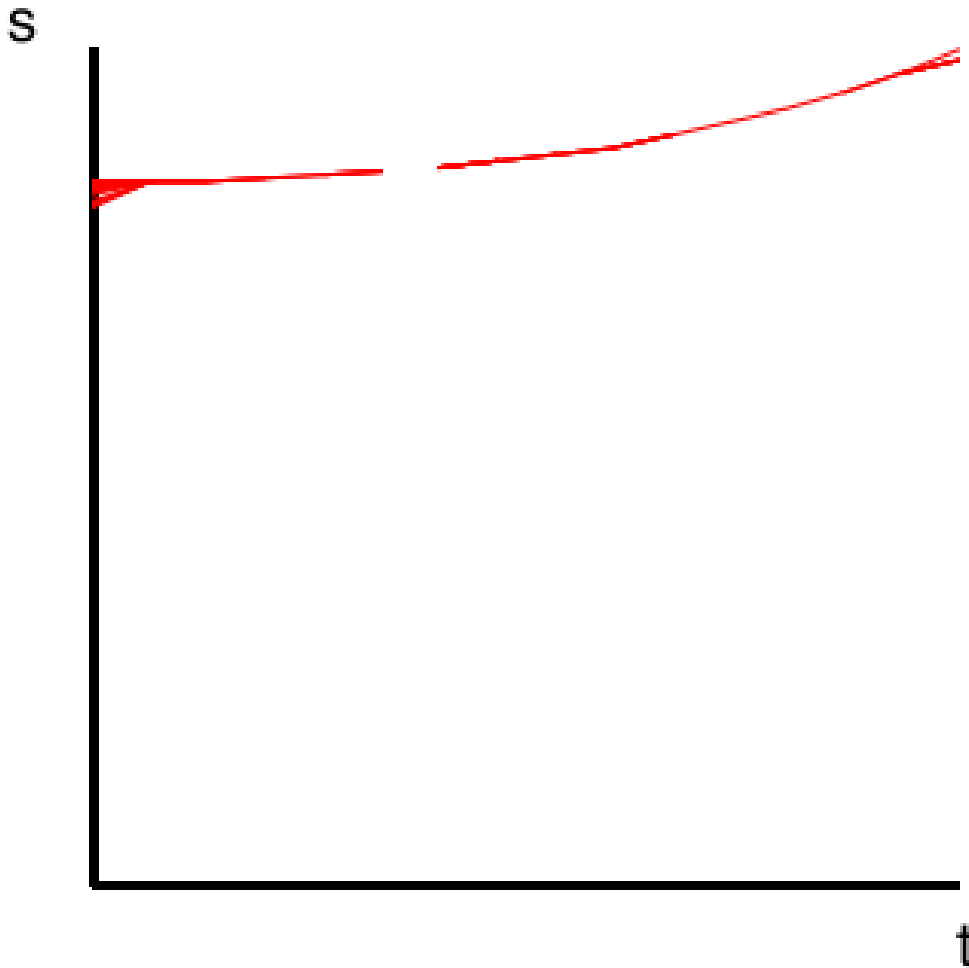
(d)

Figure 6.4: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (d) The birth fibers captured with the JCN, with the the scalar value being quantized to 256 levels, and the resolution of time being twice as large as the number of time steps. The range.



(e)

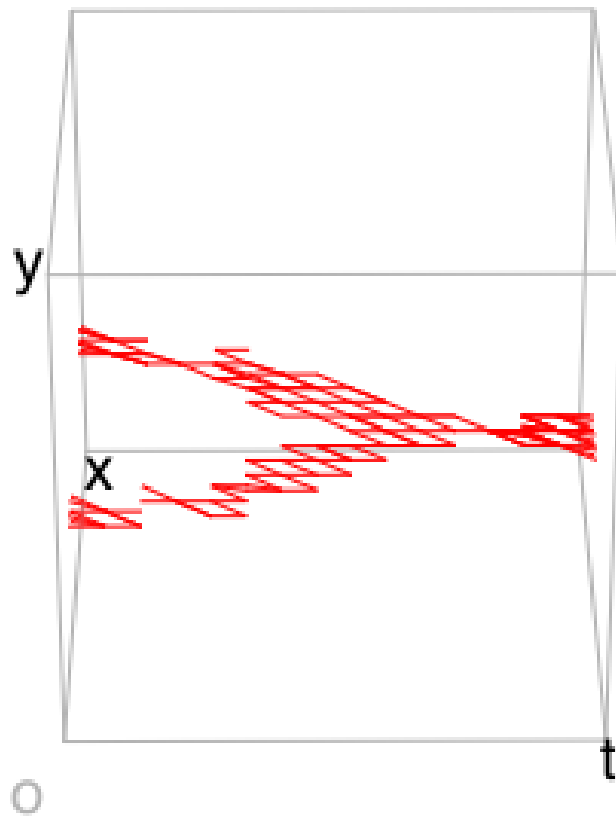
Figure 6.4: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (e) The birth fibers captured with the JCN, with the scalar value being quantized to 256 levels, and the resolution of time being four times as large as the number of time steps. The domain.



(f)

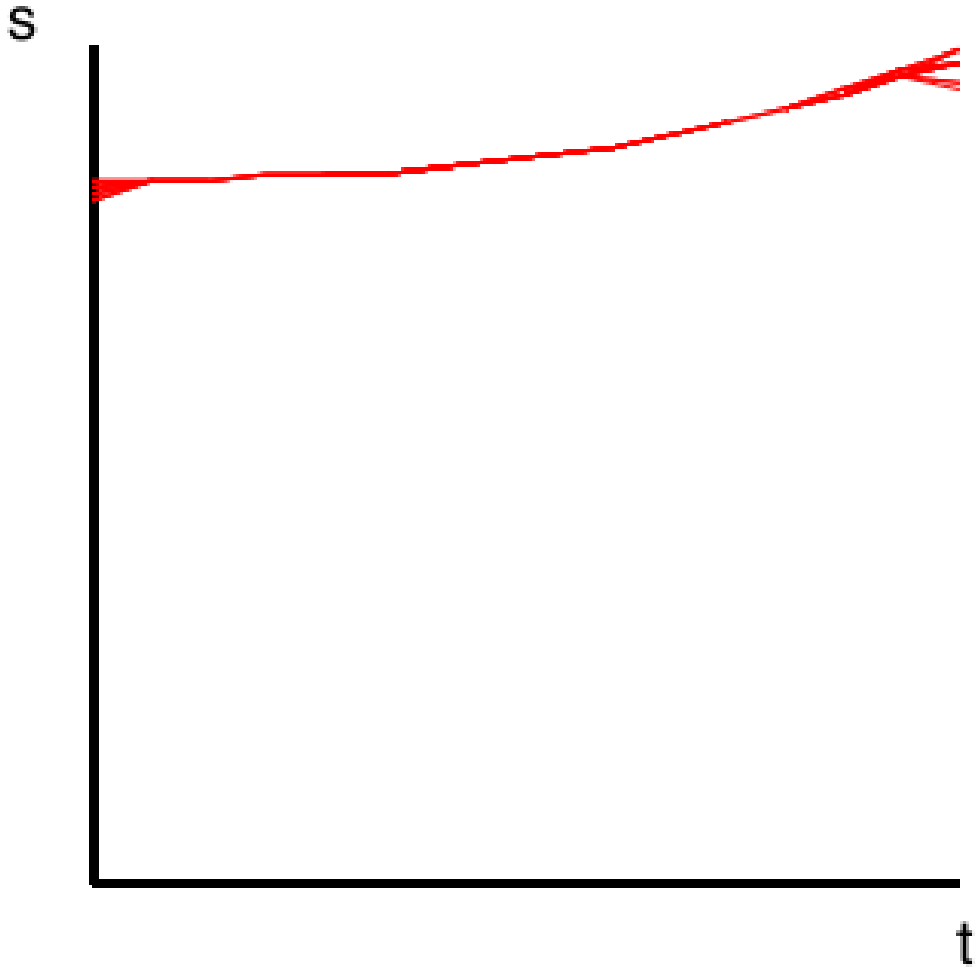
Figure 6.4: Birth fibers merging, extracted with the proposed method. Here, we vary the resolution of quantization along time. Focused view of Figure 5.3. (f) The birth fibers captured with the JCN, with the the scalar value being quantized to 256 levels, and the resolution of time being four times as large as the number of time steps. The range.





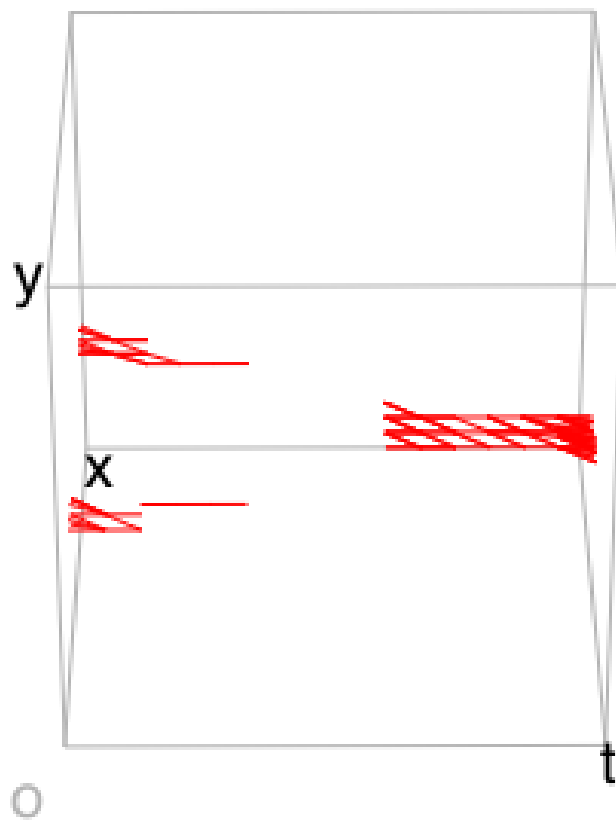
(a)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (a) The birth fibers captured with the JCN, with the scalar value being quantized to 128 levels, and the resolution of time being as large as the number of time steps. The domain.



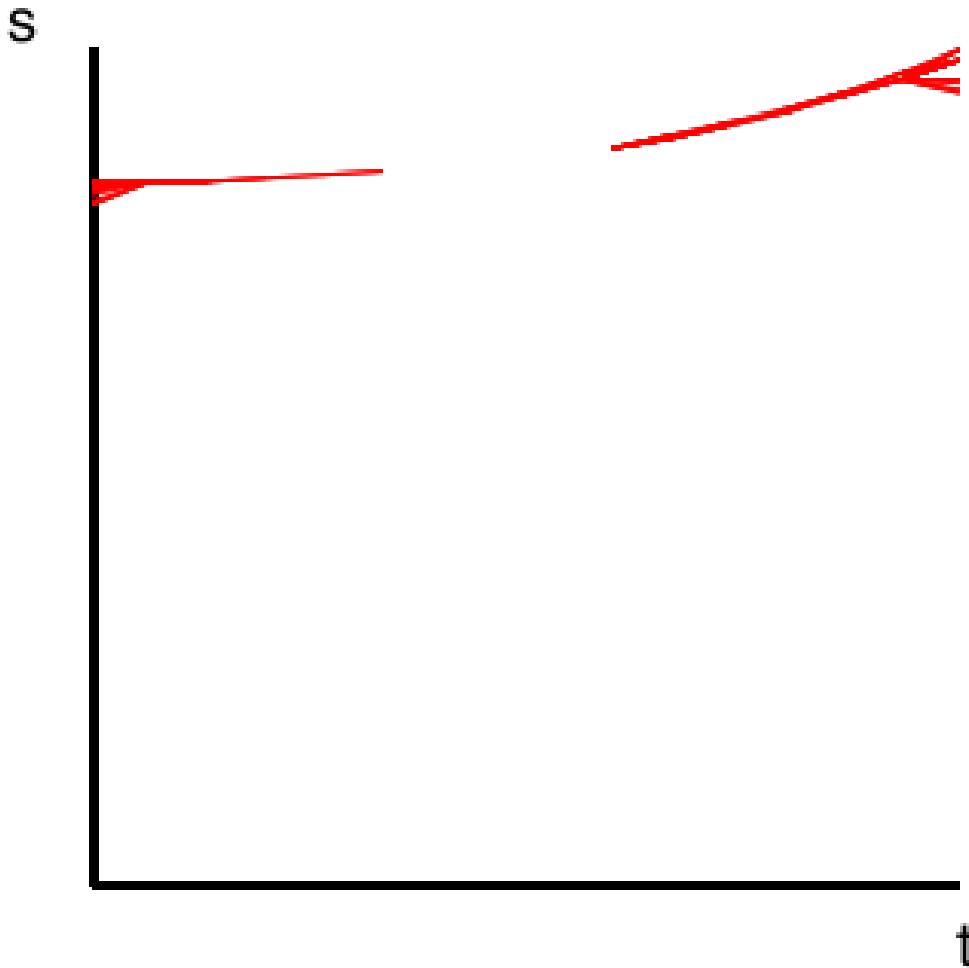
(b)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (b) The birth fibers captured with the JCN, with the same configuration as in (c): the scalar value being quantized to 128 levels, and the resolution of time being as large as the number of time steps. The range.



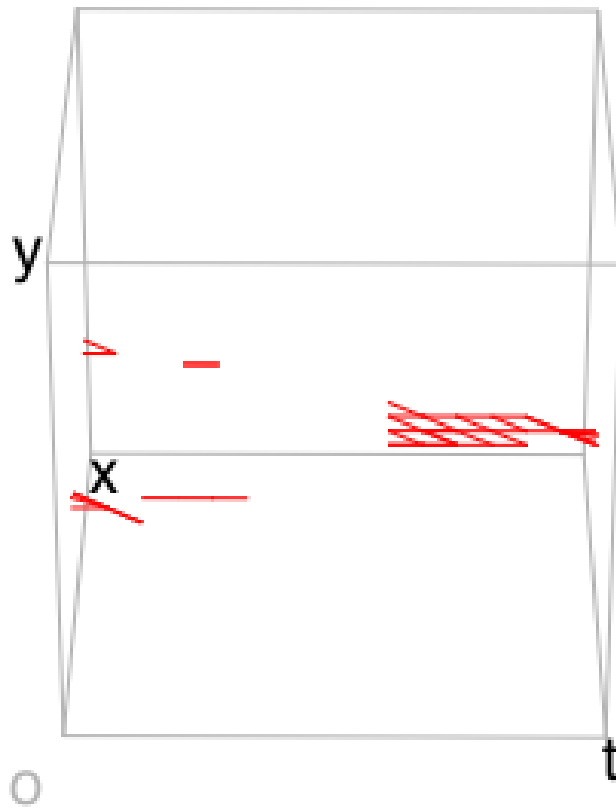
(c)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (c) The birth fibers captured with the JCN, with the scalar value being quantized to 256 levels, and the resolution of time being as large as the number of time steps. The range.



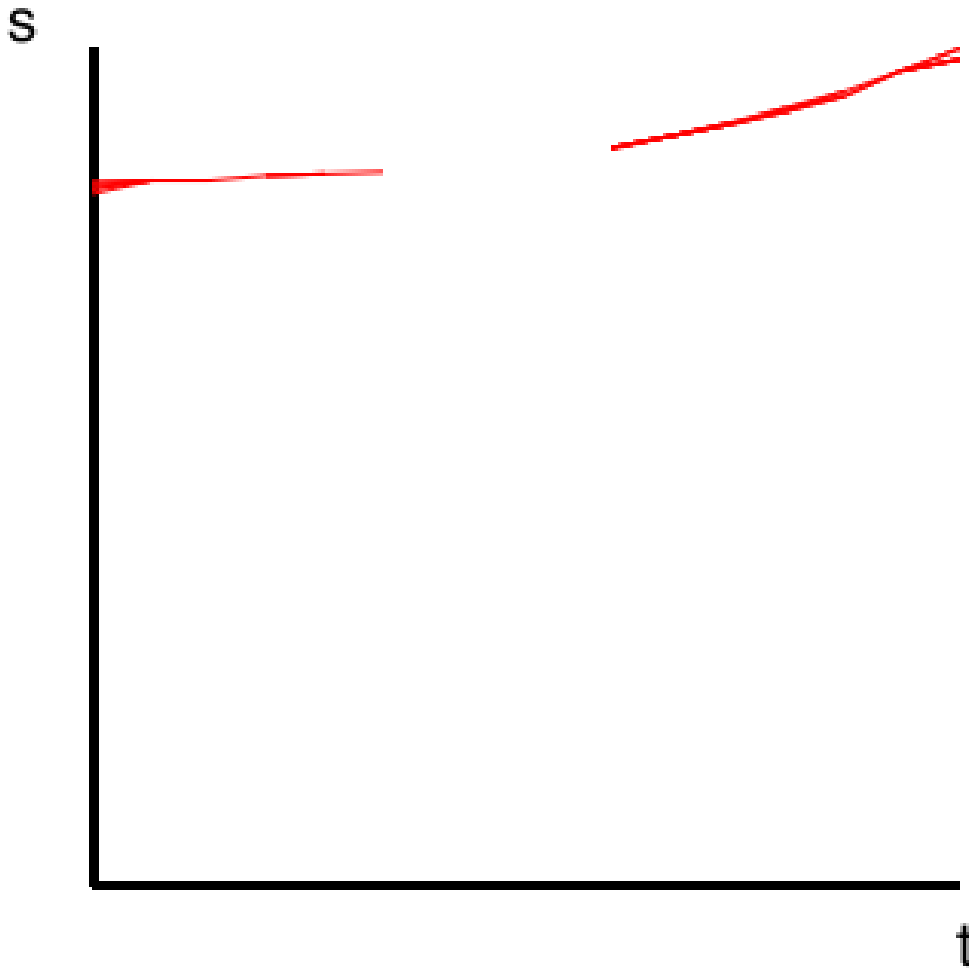
(d)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (d) The birth fibers captured with the JCN, with the same configuration as in (e): the scalar value being quantized to 256 levels, and the resolution of time being as large as the number of time steps. The range.



(e)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (e) The birth fibers captured with the JCN, with the scalar value being quantized to 512 levels, and the resolution of time being as large as the number of time steps. The domain.



(f)

Figure 6.5: Birth fibers merging, extracted with the proposed approach. Here, we vary the resolution of quantization for the scalar values. Focused view of Figure 5.3. (f) The birth fibers captured with the JCN, with the same configuration as in (e): the scalar value being quantized to 512 levels, and the resolution of time being as large as the number of time steps. The range.

point in the domain. Errors in the obtained data values naturally cause errors in extracted singular fibers. For analytic functions, small deviations in an obtained function value may give rise to fiber topology that is different from the equation. For realistic datasets, on the other hand, noise and errors in sampling may give rise to fiber topology that is different from the actual phenomenon. In fact, such faults in fiber topology may occur often, depending on the implementation of a sampling scheme. Fortunately, the errors in sampling do not affect the large-scale configuration of fibers. In other words, the deviations of fiber topology in one location does not alter the topology of fibers that are located far away. Firstly, faults in fiber topology may correspond to connected components being born, dying, splitting apart from another component, or merging with another. If not, faults are deviations in the topological shapes of fibers (including degeneracy).

In the tessellation phase, we partition the domain into a set of tetrahedra, in order to determine the interpolation of the data values. While the Delaunay tetrahedralization itself is a successful tessellation technique, different set of tetrahedra can give different interpolation of data values. This inevitably results in different fiber topology, locally. In contrast, in a global sense the topology is conserved across different interpolation results. (Indeed, the topological transitions of large fiber components are identical, for a domain with sufficiently large resolution of sampling points.) Note that this is a problem that is universal in topology-based visualization techniques.

In the graph construction phase, errors can arise while we rasterize the projection of the tetrahedra into the range. In fact, since rasterization is primarily designed to maintain the aesthetics of displayed graphics, the accuracy of pixel positions are not guaranteed. By this we mean that a sampling point may not be projected to its quantized data value, and indeed a little deviation is allowed by all of the popular implementations of rasterization. Here again, we can improve the precision by raising the resolution of quanti-

zation. However, attention must be paid to the fact that higher resolution of quantization causes capturing more details of fiber topology. Here, they may include abundant errors produced in the previous phases. However, once the tetrahedra are projected into the range and then rasterized, the graph is computed without an error since the JCN construction algorithm [CD14] is combinatorial. (It is free of round-off errors.)

When we identify singular fibers, again there is no round-off errors since the algorithm only investigates the graph structure. However, identifying the topological shape of a fiber component is done without referring to the domain, and therefore this does not give us the accurate topology for degenerate fibers. Fortunately, it was sufficient for the application in Chapter 4. Indeed, the target users were mainly interested in non-degenerate fibers. Furthermore, even the topological shapes of degenerate fibers could be visually identified with our domain view. However, detailed topology of fibers in the domain can benefit certain applications. Actually in Chapter 5, we have seen that the degeneracy of birth fibers caused uncertainty in the identifications of the local extrema's location in the domain. For accurate identification of topological shapes, we can consider further development in the future. What will be necessary is the Reeb space together with the actual configuration of fibers. This is challenging since the JCN quantizes the data values, making the individual fibers inaccessible. Another problem for analyzing fibers is that the errors in the rasterization makes it difficult to identify the exact tetrahedra that corresponds to a JCN node.

### **6.3 Miscellaneous**

In Chapter 3–5, singular fibers were extracted together with their topology. In this sense, we fully extracted the fiber types that should appear in theory. The function values were quantized but to a sufficiently detailed level.



However, the configuration of fibers in domain is limited to the closure of tetrahedra. Also, the extracted fiber topology is an approximation of the actual one. This is due to the approximation of the Reeb space by the Joint Contour Net. Nonetheless, in Chapter 4, we saw that it was sufficient for applications to analytic functions. In contrast, dealing with a more realistic data, including the simulation data in Chapter 5, the excessive amount of topological features became crucial, making it difficult to identify the actual Jacobi set captured in the JCN.

We solved the degeneracy in fiber topology by manual modification of data. Although ideally the degeneracy should be canceled automatically, manual modification proved to be useful indeed for application to mathematics. In the application to realistic datasets, the degeneracy were visualized as-is, and we did not find a practical problem.

In addition, we should consider the strategy for generalizing the dimension of the range and the domain. Indeed, our original motivation to choosing  $m = 3$  and  $n = 2$  for  $R^m \rightarrow R^n$  was to start with a simple problem to obtain knowledge for general  $m$  and  $n$ . The analysis of fiber topology via Reeb space potentially generalizes to higher dimensions, since the topological shape of fibers can be determined through the connectivity of fibers. However, to determine the topological shapes, we had to assume that the fibers were non-degenerate. The detection of topological shapes is a central problem in application to mathematics. Although our manual exploration for degenerate fibers was practical for functions  $f : R^3 \rightarrow R^2$  in our user study, such manual exploration of higher dimensional range and domain becomes difficult. Therefore, we have started a discussion to detecting degenerate fibers, and ultimately its topological shapes and types.

Furthermore, one of our users requested that we generalize the domain to manifolds, rather than a Euclidean space. Currently, the closest the interface can offer is to show the domain locally, so it can be identified as a Euclidean

space. This however increases the number of tasks for the user, and the global structure of fiber topology is difficult to obtain.

When we consider the generalization of our application of realistic datasets, two technical problems exist. The first is the generalization of the second function,  $f_2(x, y, t) = t$ . In the Jacobi set extraction, we rely on this assumption. To get rid of the assumption, we should extract the Jacobi set without calculation errors. Ultimately, if we were to extract the Reeb space without quantizations of function values, then it should also be free of this assumption. The second is the generalization of the dimensions of the range and the domain. Although the simplification of the Reeb space via the Reeb skeleton [CCDGb] generalizes, its relation with Jacobi set is not discussed.

# Chapter 7

## Conclusion and Future Work

This thesis presented an approach to extract, identify, and explore multivariate data, as multi-fields and time-varying scalar fields. The exploration of the distribution of data values was achieved by characterizing the topological transitions of the fibers with singular fibers, and showing them together with the connectivity between fibers.

Indeed in Chapter 4, which explained the application of the proposed approach to facilitating tasks among mathematicians, participants confirmed that the interface can provide new insights on fiber topology in functions. Before this work, visualizing the singularity of fibers and the connectivity between fibers had been a technique primarily for data analysts. Interestingly, however, the work revealed that also mathematicians can find benefits. In the near future, studies should be sought to obtain new mathematical findings. Ultimately, such findings may offer new tools to the visualization community.

In addition, the information of singular fibers with their connectivity was also advantageous in analyzing real datasets. In this work, Chapter 5 sought an application of multivariate fiber topology to real datasets as time-varying scalar fields. The advantage of the present approach was to respect the continuity of local extrema over time, by setting the time as an auxiliary function. Such continuity enabled reducing the visual clutters in the results

and distinguishing the size of physical phenomena, which were vortices in 2D fluids.

However, in both situations, the quality of feature extraction relied on the selection of quantization level. Especially, the identification of the exact locations for local extrema could not be achieved in Chapter 5, since the quantization introduced degeneracies of fibers. For extracting topology in more details, and encoding it, a singular fiber extraction method without quantizing data values should be necessary. In addition, ideally all the topological patterns of the Reeb space in Figure 5.2 should be simplified with a single coherent strategy. Here, the quantization of function values prevented the extraction of the details of topology such as the position of a cusp in the domain and its image in the range. Therefore, the present work has focused only on the eye pattern.

Extension of the approach should also be sought. Such extension includes handling high-dimensional manifolds. In addition, visualization of the Reeb space such as the Reeb space view in Chapter 4 should be achieved so that it will effectively inform the connectivity of complicated data.

# Bibliography

- [ADM92] Y. Arnaud, M. Desbois, and J. Maizi. Automatic tracking and characterization of African convective systems on meteosat pictures. *Journal of Applied Meteorology*, 31(5):443–453, 1992.
- [Bal82] D. Ballard. *Computer Vision*. Englewood, N.J.: Prentice Hall, Inc., 1982.
- [BBD<sup>+</sup>07] P.-T. Bremer, E. M. Bringa, M. A. Duchaineau, A. G. Gyulassy, D. Laney, A. Mascarenhas, and V. Pascucci. Topological feature extraction and tracking. *Journal of Physics: Conference Series*, 78:012007, 2007.
- [BDSW13] A. Biswas, S. Dutta, H.-W. Shen, and J. Woodring. An information-aware framework for exploring multivariate data sets. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2683–2692, 2013.
- [BHEP04] P.-T. Bremer, B. Hamann, H. Edelsbrunner, and V. Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.
- [BKSW07] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Computational Geometry*, 36(3):215–236, 2007.

- [BPS97] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *Proceedings of IEEE Visualization '97*, pages 167–173, 1997.
- [BR63] R. L. Boyell and H. Ruston. Hybrid techniques for real-time radar simulation. In *Proceedings of the 1963 Fall Joint Computer Conference, AFIPS '63*, pages 445–458. ACM, 1963.
- [BW08] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.
- [BWN<sup>+</sup>13] H. Bhatia, B. Wang, G. Norgard, V. Pascucci, and P.-T. Bremer. Local, smooth and consistent Jacobi set simplification. Technical Report CoRR abs/1307.7752, arXiv, 2013.
- [CCDGa] A. Chattopadhyay, H. Carr, D. Duke, and Z. Geng. Multivariate topology simplification and measure persistence. *Computational Geometry*. Under review.
- [CCDGb] A. Chattopadhyay, H. Carr, D. Duke, and Z. Geng. Simplifying multivariate topology. In *Computer Graphics and Visual Computing, CGVC 2014*, pages 10–11.
- [CD14] H. Carr and D. Duke. Joint contour nets. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1100–1113, 2014.
- [CDD06] H. Carr, B. Duffy, and B. Denby. On histograms and isosurface statistics. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1259–1266, 2006.

- [CS03] H. Carr and J. Snoeyink. Path seeds and flexible isosurfaces – using topology for exploratory visualization. In *Proceedings of the Symposium on Data Visualisation*, pages 49–58, 2003.
- [CS09] H. Carr and J. Snoeyink. Representing interpolant topology for contour tree computation. In H.-C. Hege, K. Polthier, and G. Scheuermann, editors, *Topology-Based Methods in Visualization II*, Mathematics and Visualization, pages 59–73. Springer Berlin Heidelberg, 2009.
- [CSA03] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Computational Geometry*, 24(2):75–94, 2003.
- [CSvdP04] H. Carr, J. Snoeyink, and M. van de Panne. Simplifying flexible isosurfaces using local geometric measures. In *Proceedings of IEEE Visualization 2004*, pages 497–504, 2004.
- [DCH88] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. *ACM Computer Graphics*, 22(4):65–74, 1988.
- [DCK<sup>+</sup>12] D. Duke, H. Carr, A. Knoll, N. Schunck, H. A. Nam, and A. Staszczak. Visualizing nuclear scission through a multi-field extension of topological analysis. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2033–2040, 2012.
- [DOL03] M. C. F. De Oliveira and H. Levkowitz. From visual data exploration to visual data mining: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):378–394, 2003.
- [Ede09] H. Edelsbrunner. *Computational Topology*. American Mathematical Society, 2009.

- [EH04] H. Edelsbrunner and J. Harer. Jacobi sets of multiple Morse functions. In *Foundations of Computational Mathematics*, pages 37–57, 2004.
- [EHMP04] H. Edelsbrunner, J. Harer, A. Mascarenhas, and V. Pascucci. Time-varying Reeb graphs for continuous space-time data. In *Proceedings of the 20th Annual Symposium on Computational Geometry*, SCG '04, pages 366–372. ACM, 2004.
- [EHP08] H. Edelsbrunner, J. Harer, and A. K. Patel. Reeb spaces of piecewise linear mappings. In *Proceedings of the 24th Annual Symposium on Computational Geometry*, pages 242–250, 2008.
- [EHZ03] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete & Computational Geometry*, 30(1):87–107, 2003.
- [FH09] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.
- [FM67] S. Freeman and S. P. Morse. On searching a contour map for a given terrain elevation profile. *Journal of the Franklin Institute*, 284(1):1–25, 1967.
- [FOTT08] I. Fujishiro, R. Otsuka, S. Takahashi, and Y. Takeshima. T-map: A topological approach to visual exploration of time-varying volume data. In J. Labarta, K. Joe, and T. Sato, editors, *High-Performance Computing*, volume 4759 of *Springer Lecture Notes in Computer Science*, pages 176–190. 2008.
- [FR91] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.



- [FTAT00] I. Fujishiro, Y. Takeshima, T. Azuma, and S. Takahashi. Volume data mining using 3D field topology analysis. *IEEE Computer Graphics and Applications*, 20:46–51, 2000.
- [GBPW10] S. Gerber, P.-T. Bremer, V. Pascucci, and R. Whitaker. Visual exploration of high dimensional scalar functions. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1271–1280, 2010.
- [GDS<sup>+</sup>06] F. Gygi, E. W. Draeger, M. Schulz, B. R. de Supinski, J. A. Gunnels, V. Austel, J. C. Sexton, F. Franchetti, S. Kral, C. W. Ueberhuber, and J. Lorenz. Large-scale electronic structure calculations of high-Z metals on the BlueGene/L platform. In *Proceedings of the 2006 ACM/IEEE Conference on Supercomputing*, SC '06, New York, NY, USA, 2006. ACM.
- [GK11] D. T. Gay and R. C. Kirby. Fiber-connected, indefinite Morse 2-functions on connected  $n$ -manifolds. *Proceedings of the National Academy of Sciences*, 108(20):8122–8125, 2011.
- [HM78] A. Hasegawa and K. Mima. Pseudo-three-dimensional turbulence in magnetized nonuniform plasma. *Physics of Fluids*, 21(87), 1978.
- [HS13] J. T. Hiratuka and O. Saeki. Triangulating Stein factorizations of generic maps and Euler characteristic formulas. *RIMS Kokyuroku Bessatsu*, B38:61–89, 2013.
- [HSCS11] C. Heine, D. Schneider, H. Carr, and G. Scheuermann. Drawing contour trees in the plane. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1599–1611, 2011.

- [HSKK01] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '01*, pages 203–212. ACM, 2001.
- [IS09] K. Ikegami and O. Saeki. Cobordism of Morse maps and its application to map germs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 147:235–254, 2009.
- [JSW03] G. Ji, H.-W. Shen, and R. Wenger. Volume tracking using higher dimensional isosurfacing. In *Proceedings of IEEE Visualization 2003*, pages 209–216, 2003.
- [KKH01] J. Kniss, G. Kindlmann, and C. Hansen. Interactive volume rendering using multi-dimensional transfer functions and direct manipulation widgets. In *Proceedings of IEEE Visualization 2001*, pages 255–262, 2001.
- [KKSA10] Y. Kiwamoto, Y. Kawai, Y. Soga, and J. Aoki. Turbulence and structure formation associated with vortex dynamics in non-neutral plasma flow. *Plasma and Fusion Research*, 5:S2002–S2002, 2010.
- [KRS03] L. Kettner, J. Rossignac, and J. Snoeyink. The Safari interface for visualizing time-dependent volume data using iso-surfaces and contour spectra. *Computational Geometry*, 25(1):97–116, 2003.
- [LBM<sup>+</sup>06] D. Laney, P.-T. Bremer, A. Mascarenhas, P. Miller, and V. Pascucci. Understanding the structure of the turbulent mixing

- layer in hydrodynamic instabilities. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):1053–1060, 2006.
- [LC87] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM Computer Graphics*, 21(4):163–169, 1987.
- [Lev85] H. Levine. *Classifying Immersions into  $R^4$  over Stable Maps of 3-manifolds into  $R^2$* , volume 1157 of *Springer Lecture Notes in Mathematics*. 1985.
- [LHL<sup>+</sup>98] Z. Lin, T. S. Hahm, W. W. Lee, W. M. Tang, and R. B. White. Turbulent transport reduction by zonal flows: Massively parallel simulations. *Science*, 281(5384):1835–1837, 1998.
- [LT10] D. J. Lehmann and H. Theisel. Discontinuities in continuous scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1291–1300, 2010.
- [MCC<sup>+</sup>99] A. A. Mirin, R. H. Cohen, B. C. Curtis, W. P. Dannevik, A. M. Dimits, M. A. Duchaineau, D. E. Eliason, D. R. Schikore, S. E. Anderson, D. H. Porter, P. R. Woodward, L. J. Shieh, and S. W. White. Very high resolution simulation of compressible turbulence on the IBM-SP system. In *Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*, 70, 1999.
- [MS09] A. Mascarenhas and J. Snoeyink. Isocontour based visualization of time-varying scalar fields. In T. Möller, B. Hamann, and R. D. Russell, editors, *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization, pages 41–68. Springer Berlin Heidelberg, 2009.

- [NB13] G. Norgard and P.-T. Bremer. Ridge-valley graphs: Combinatorial ridge detection using Jacobi sets. *Computer Aided Geometric Design*, 30(6):597–608, 2013.
- [Nie03] G. M. Nielson. On marching cubes. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):341–351, 2003.
- [OHJ<sup>+</sup>11] P. Oesterling, C. Heine, H. Jänicke, G. Scheuermann, and G. Heyer. Visualization of high-dimensional point clouds using their density distribution’s topology. *IEEE Transactions on Visualization and Computer Graphics*, 17(11):1547–1559, 2011.
- [PCM02] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Proceedings of IEEE Visualization 2002*, pages 187–194, 2002.
- [PPF<sup>+</sup>09] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser. On the way towards topology-based visualization of unsteady flow – the state of the art. technical report. In *Eurographics 2010 STAR Reports*, pages 137–154, 2009.
- [PSBM07] V. Pascucci, G. Scorzelli, P.-T. Bremer, and A. Mascarenhas. Robust on-line computation of Reeb graphs: Simplicity and speed. *ACM Transactions on Graphics*, 26(3):58, 2007.
- [Ree46] G. Reeb. Sur les points singuliers d’une forme de pfaff complètement intégrable ou d’une fonction numérique. *Comptes Rendus de l’Académie des Sciences de Paris*, 222:847–849, 1946.
- [Sae04] O. Saeki. *Topology of Singular Fibers of Differentiable Maps*, volume 1854 of *Springer Lecture Notes in Mathematics*. 2004.

- [SB06] B.-S. Sohn and C. Bajaj. Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics*, 12(1):14–25, 2006.
- [SC86] J. K. Sircar and J. A. Cerbrian. Application of image processing techniques to the automated labelling of raster digitized contours. In *Proceedings of the 2nd International ACM Symposium on Spatial Data Handling*, pages 171–184, 1986.
- [SDHH12] S. Shaffi, S. E. Dillard, M. Hlawitschka, and B. Hamann. The topological effects of smoothing. *IEEE Transactions on Visualization and Computer Graphics*, 18(1):160–172, 2012.
- [SN11] N. Suthambhara and V. Natarajan. Simplification of Jacobi sets. In *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, pages 91–102. 2011.
- [SSC<sup>+</sup>] D. Sakurai, O. Saeki, H. Carr, H.-Y. Wu, T. Yamamoto, D. Duke, and S. Takahashi. Interactive visualization for singular fibers of functions  $f : R^3 \rightarrow R^2$ . Submitted to *IEEE Scientific Visualization 2015 (SciVis 2015)*.
- [SSZC94] R. Samtaney, D. Silver, N. Zabusky, and J. Cao. Visualizing features and tracking their evolution. *Computer*, 27(7):20–27, 1994.
- [STS<sup>+</sup>14] O. Saeki, S. Takahashi, D. Sakurai, H.-Y. Wu, K. Kikuchi, H. Carr, D. Duke, and T. Yamamoto. Visualizing multivariate data using singularity theory. In M. Wakayama, R. S. Anderssen, J. Cheng, Y. Fukumoto, R. McKibbin, K. Polthier, T. Takagi, and K.-C. Toh, editors, *The Impact of Applications*

*on Mathematics, Proceedings of the Forum of Mathematics for Industry 2013*, pages 51–65. Springer, 2014.

- [SW97] D. Silver and X. Wang. Tracking and visualizing turbulent 3D features. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):129–141, 1997.
- [SW98] D. Silver and X. Wang. Tracking scalar features in unstructured data sets. In *Proceedings of IEEE Visualization '98*, pages 79–86, 1998.
- [SWC<sup>+</sup>08] D. Schneider, A. Wiebel, H. Carr, M. Hlawitschka, and G. Scheuermann. Interactive comparison of scalar fields based on largest contours with applications to flow visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1475–1482, 2008.
- [SY14] O. Saeki and T. Yamamoto. Singular fibers of stable maps of 3-manifolds with boundary into surfaces and their applications. *preprint*, 2014.
- [TFO09] S. Takahashi, I. Fujishiro, and M. Okada. Applying manifold learning to plotting approximate contour trees. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1185–1192, 2009.
- [TFT05] S. Takahashi, I. Fujishiro, and Y. Takeshima. Interval volume decomposer: A topological approach to volume traversal. In *Proceedings of Conference on Visualization and Data Analysis '05*, pages 103–114, 2005.
- [TFTH13] Y. Takeshima, I. Fujishiro, S. Takahashi, and T. Hayase. A topologically-enhanced juxtaposition tool for hybrid wind tun-

- nel. In *Proceedings of 2013 IEEE Pacific Visualization Symposium (Pacific Vis)*, pages 113–120, 2013.
- [TIS<sup>+</sup>95] 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.
- [Tre00] L. A. Treinish. Multi-resolution visualization techniques for nested weather models. In *Proceedings of IEEE Visualization 2000*, pages 513–516, 2000.
- [TTF04] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):24–49, 2004.
- [TTFN05] Y. Takeshima, S. Takahashi, I. Fujishiro, and G. M. Nielson. Introducing topological attributes for objective-based visualization of simulated datasets. In A. E. Kaufman, K. Mueller, E. Gröller, D.W. Fellner, T. Möller, and S.N. Spencer, editors, *Proceedings of Volume Graphics 2005*, pages 137–145. Eurographics Association, 2005.
- [TV98] S. P. Tarasov and M. N. Vyalyi. Construction of contour trees in 3D in  $O(N \log N)$  steps. In *Proceedings of the 14th Annual Symposium on Computational Geometry*, pages 68–75. ACM, 1998.
- [vKvOB<sup>+</sup>97] M. J. van Kreveld, R. van Oostrum, C. L. Bajaj, V. Pascucci, and D. Schikore. Contour trees and small seed sets for isosurface traversal. In *Proceedings of the 13th Annual Symposium on Computational Geometry*, pages 212–220. ACM, 1997.

- [VV92] J. Villasenor and A. Vincent. An algorithm for space recognition and time tracking of vorticity tubes in turbulence. *CVGIP: Image Understanding*, 55(1):27–35, 1992.
- [WB97] P. C. Wong and R. D. Bergeron. 30 years of multidimensional multivariate visualization. *Scientific Visualization Overviews, Methodologies, and Techniques*, pages 3–33, 1997.
- [WBP07] G. H. Weber, P.-T. Bremer, and V. Pascucci. Topological landscapes: A terrain metaphor for scientific data. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1416–1423, 2007.
- [WDC<sup>+</sup>07] G. H. Weber, S. E. Dillard, H. Carr, V. Pascucci, and B. Hamann. Topology-controlled volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(2):330–341, 2007.
- [WSH03] G.H. Weber, G. Scheuermann, and B. Hamann. Detecting critical regions in scalar fields. In *Proceedings of the Symposium on Data Visualisation 2003*, VISSYM '03, pages 85–94. Eurographics Association, 2003.
- [WSHH02] G. H. Weber, G. Scheuermann, H. Hagen, and B. Hamann. Exploring scalar fields using critical isovalues. In *Proceedings of the IEEE Visualization 2002*, pages 171–178, 2002.