Article:
Sakurai, D, Saeki, O, Carr, H et al. (4 more authors) (2016) Interactive Visualization for Singular Fibers of Functions f : R3 → R2. IEEE Transactions on Visualization and Computer Graphics, 22 (1). pp. 945-954. ISSN 1077-2626

https://doi.org/10.1109/TVCG.2015.2467433

Interactive Visualization for Singular Fibers of Functions $f : R^3 \to R^2$

Daisuke Sakurai, *Member, IEEE*, Osamu Saeki, Hamish Carr, *Member, IEEE*, Hsiang-Yun Wu, *Member, IEEE*, Takahiro Yamamoto, David Duke, *Member, IEEE*, and Shigeo Takahashi, *Member, IEEE*

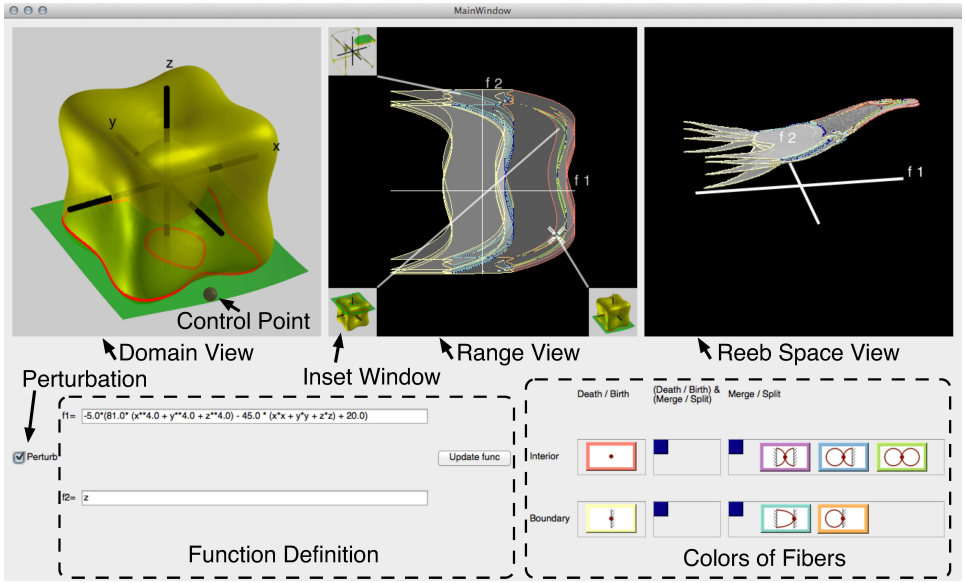


Fig. 1. Singular fibers of the tangle cube function $f(x,y,z) = (-x^4 - y^4 - z^4 + 5(x^2 + y^2 + z^2) - 10, z)$ after interactive perturbation. The domain view on the left visualizes isosurfaces of individual axis function in yellow and green with their intersection along the current fiber in red. In the center, the range view shows 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 view are thumbnails of the domain space that pop up when a value is selected. The Reeb space view shows the connectivity of connected components in the fiber, constructed by using a 3D layout of the Joint Contour Net.

Abstract— Scalar topology in the form of Morse theory has provided computational tools that analyze and visualize data from scientific and engineering tasks. Contracting isocontours to single points encapsulates variations in isocontour connectivity in the Reeb graph. For multivariate data, isocontours generalize to fibers—inverse images of points in the range, and this area is therefore known as fiber topology. However, fiber topology is less fully developed than Morse theory, and current efforts rely on manual visualizations. This paper presents how to accelerate and semi-automate this task through an interface for visualizing fiber singularities of multivariate functions $\boldsymbol{R}^3 \to \boldsymbol{R}^2$. This interface exploits existing conventions of fiber topology, but also introduces a 3D view based on the extension of Reeb graphs to Reeb spaces. Using the Joint Contour Net, a quantized approximation of the Reeb space, this accelerates topological visualization and permits online perturbation to reduce or remove degeneracies in functions under study. Validation of the interface is performed by assessing whether the interface supports the mathematical workflow both of experts and of less experienced mathematicians.

Index Terms—Singular fibers, fiber topology, mathematical visualization, design study.

- *Daisuke Sakurai is with the University of Tokyo and Japan Atomic Energy Agency, Kashiwa, Japan. E-mail: d.sakurai@computer.org.*
- *Osamu Saeki is with Kyushu University, Fukuoka, Japan. E-mail: saeki@imi.kyushu-u.ac.jp.*
- *Hamish Carr and David Duke are with the University of Leeds, Leeds, UK. E-mail: {h.carr,d.j.duke}@leeds.ac.uk.*
- *Hsiang-Yun Wu is with Keio University, Yokohama, Japan. E-mail: hsiang.yun.wu@computer.org.*
- *Takahiro Yamamoto is with Kyushu Sangyo University, Fukuoka, Japan. E-mail: yama.t@ip.kyusan-u.ac.jp.*
- *Shigeo Takahashi is with the University of Aizu, Aizu-Wakamatsu, Japan. E-mail: takahashis@compuer.org.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication xx Aug. 2015; date of current version 25 Oct. 2015. For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.

1 Introduction

It is often assumed that mathematics supports visualization with concepts and equations to build upon. Yet there are also problems where computation can support 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) even for multi-fields, in order to form the *singular fiber theory*. In the advancements of this theory, manual visualization has supported obtaining and communicating significant results. Thus, semi-automation of this task further facilitates both experts and learners to investigate phenomena that are too complex for manual visualization. As with the scalar case, topology of fibers in multi-fields is

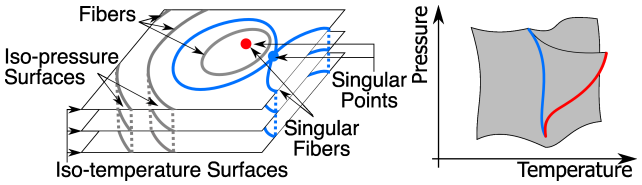


Fig. 2. Singular fiber of a function $f : \boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$ (on the left) with its Reeb space (on the right). Isosurfaces of temperature are horizontal planes, while isosurfaces of pressure are curved sheets perpendicular to the planes. Fibers (i.e. inverse images) of the function are curves obtained as intersections between isosurfaces of temperature and pressure. The Reeb space represents the connectivity of such fibers in terms of temperature and pressure.

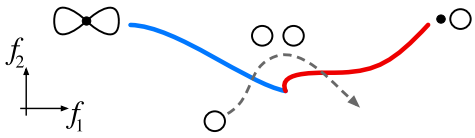


Fig. 3. A typical hand-drawn fiber visualization: red and blue curves indicate singular values in the range. The topological shapes of fiber components changes as one walks through the dashed path. At the blue curve, a fiber component splits into two. At the red curve, one fiber component becomes a point and then disappears. The topological shapes of fibers are represented by icons.

now being applied to data analysis and visualization [6]. Here again, manual visualization helps communicate the mathematical results to non-specialists, and develop the intuition of novices to the field. This suggests that there exists considerable advantage in developing visualization systems for that task.

Although work exists on extracting topological features from multi-fields, an effective visualization system for supporting the mathematical study is yet challenging to develop. Indeed, such topological features are more complex than those in the scalar case, even for the simplest case where 3-manifolds are transformed to 2-manifolds.

This paper therefore reports a user interface (Figure 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 $f : \boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$. Interestingly enough, the available *Reeb space* analysis [6], which is the extension of the familiar Reeb graph to multi-fields, provides necessary information about the fiber topology. Our technical contribution is firstly to bridge the gap between the mathematical basement presented in [30] and the associated computational algorithm. For that purpose, we will describe the algorithmic details here in this paper. In addition, we let the user perturb the function intuitively to resolve duplicated singularities into simple ones. Another contribution lies in the visualization of Reeb space in 3D.

We start with the mathematical preliminaries in Section 2. Section 3 then discusses the relevant prior results in the visualization literature. Then Section 4 characterizes the mathematical workflow in singular fiber theory, and identifies bottlenecks where machine visualization can accelerate tasks. Section 5 discusses the design of the application, and Section 6 the new techniques required. Finally, Section 7 gives some implementation details, Section 8 discusses the use of the interface, and Section 9 summarizes and discusses future directions.

2 Singular Fibers

Singular fiber theory [29] 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 $f : \boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$, as this is challenging, but can still be displayed in a 3D interface, and develops intuition for higher-dimensional cases. Readers can refer to [30] for a formal introduction to the mathematical backgrounds.

We assume a function $f : \boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$ 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 $\boldsymbol{R}^2$, 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 2. Here, the planes are isosurfaces of temperature, while the stacked curves belong to isosurfaces of pressure. For a particular pressure and temperature, we find the fiber as intersections between the pressure isosurface and temperature isosurface. Since these isosurfaces are 2-manifolds, most fibers are 1-manifolds embedded in $\boldsymbol{R}^3$, 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 2, 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.

Fiber components whose topological shape is *not* an arc or loop are said to be *singular*, and fibers with any singular component are also singular. Figure 2 indicates such singular fiber components in red and blue, and Figure 4 lists more examples. Any fiber consisting only of arcs or loops is 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. In the scalar case, one intuitively understood the topological changes while sweeping through the 1D range. A similar view can be employed to understand the multi-field. As in Figure 3, consider a path in the range. The topological shape evolves smoothly along the path, while changing from a single loop to two loops at a *singular value* on the blue curve where the fiber takes on a figure-8 shape, and then into a single loop again through the red curve where one fiber component shrinks to a point. Here, the types of these singular fibers correspond to type V and type I in Figure 4, respectively.

Returning briefly to scalar fields, we see that the critical values divide the 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 in [9] 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), a region where all topological shapes consist of one loop must necessarily 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 may break into two, and the critical point is the border between the one contour and two contours. At some higher isovalue, the two contours may either reconnect, or one may disappear entirely at a local maximum. In Figure 3, the red curve carries the semantics of a local extremum (i.e. extinction), while the blue curve carries the semantics of a saddle point (separation).

We note that the set of singular points is the *Jacobi set* of a function [12]. As described earlier, in scalar fields, the connectivity of fiber components across different function values was represented as the Reeb graph. This can be naturally extended to the *Reeb space* in the multi-field case, which is defined as a quotient space by identifying each fiber component with a point. It provides an intuitive representation of the connectivity of fibers, as in Figure 2. We also note that the Joint Contour Net (JCN) [6] is a discrete representation of the Reeb space, and we will employ this representation in our approach.

Singular fiber theory [29] 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

Type	Color	Topological Change	# Fiber Components	# Boundary Points	Description
0a			$1 \to 1 \to 1$	$0 \to 0 \to 0$	Regular interior fiber component.
0b			$1 \to 1 \to 1$	$2 \to 2 \to 2$	Regular boundary fiber component.
I			$0 \to 1 \to 1$	$0 \to 0 \to 0$	Birth of an interior fiber component.
II			$1 \to 1 \to 1$	$0 \to 1 \to 2$	An interior fiber component meets the boundary.
III			$0 \to 1 \to 1$	$0 \to 1 \to 2$	Birth of a boundary fiber component.
IV			$1 \to 1 \to 2$	$2 \to 3 \to 4$	A boundary fiber component splits at boundary.
V			$1 \to 1 \to 2$	$0 \to 0 \to 0$	An interior fiber component splits.
VI			$1 \to 1 \to 2$	$2 \to 2 \to 2$	A boundary fiber component splits in the interior.
VII			$2 \to 1 \to 2$	$4 \to 4 \to 4$	Two boundary fiber components meet and re-pair.

Fig. 4. Topology of non-degenerate fiber components, after Saeki and Yamamoto [31]. For $f : \boldsymbol{R}^3 \to \boldsymbol{R}^2$, topological changes to fibers must be one of these possibilities or their converses, barring degeneracy. These are the same changes as for isolines in $f : \boldsymbol{R}^2 \to \boldsymbol{R}^1$ – i.e. scalar critical points in a scalar field. However, since there are *two* range variables, these changes occur across one dimensional boundaries in the range rather than in a strict linear sequence, as shown in Figure 3.

domain without boundary, and it is only recently that it has been extended to deal with behavior at the boundary of a finite domain [31]. 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 current paper.

This classification is shown in Figure 4, 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. This is analogous to contours in the scalar case.

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 [29, 31], 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 5.

The list in Figure 4 is not always easy to interpret in practice, as multiple topological changes can occur in a single fiber, especially in degenerate cases. 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 in order to visualize the mathematical classification of fibers.

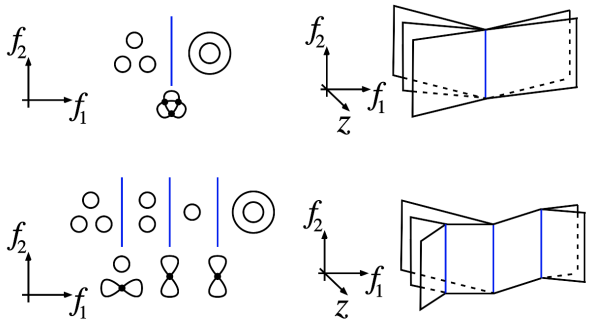


Fig. 5. A degenerate fiber in a Reeb space. On the left, topological shapes of fibers shown in the range. On the right, the Reeb space, expanded along a virtual dimension z for the visual clarity. Above, three fiber components merge simultaneously at three singular points, along the singular values indicated by the blue line. Below, a perturbation to the function resolves the component to non-degenerate ones.

3 Related Work

In scientific visualization, topological analysis was introduced through the use of the contour tree as a graph representing the topological complexity of isosurfaces [2]. Later, contour trees were used for semantic segmentation of the entire 3D scalar field [8]. Kettner et al. [22] extended it to portray topological complexity in a time-varying dataset by counting connected components as isovalue and time varied, and then presented the information as a color map for visual analysis. While their interface were based on analyzing the number of fiber components, we identify and visualize the singularities in fiber topology. Actually, a time-varying scalar field [10, 22, 13] becomes a multi-field by choosing the time as an auxiliary function, and the visualization community obtained a classification of fibers [10] similar to Figure 4.

More generally, algorithms have been reported for constructing contour trees [9] and Reeb graphs [28]. These were then enhanced with additional information such as isosurface genus [27] and spatial embeddings [35], and employed as visual metaphors of the given datasets [36]. Moreover, these approaches have also been used to reduce very high dimensional data to two or three for visualization [34, 26, 17].

Multi-fields are usually more difficult to analyze than scalar fields, so fewer effective techniques are known [16]. Bachthaler and Weiskopf [1] extended conventional scatterplots to the continuous version, projecting smooth density within the domain space onto the range space. Lehmann and Theisel [23] extended this work by identifying discontinuities as important features, and it is now clear that these features contains the singular values of multi-fields. Alternatively, Huettenberger et al. [20, 21] extended scalar topology analysis to multi-fields by adapting the concept of Pareto optimality. In practice, the optimal points form a subset of the Jacobi set [20]. Another effective visualization technique is the *fiber surface* [7], which extracts a set of fibers as surfaces, allowing the user to specify it as curves in the range.

An alternate approach to multi-fields analyzes contour trees of individual fields, then compares the features of each [32], but this is not the

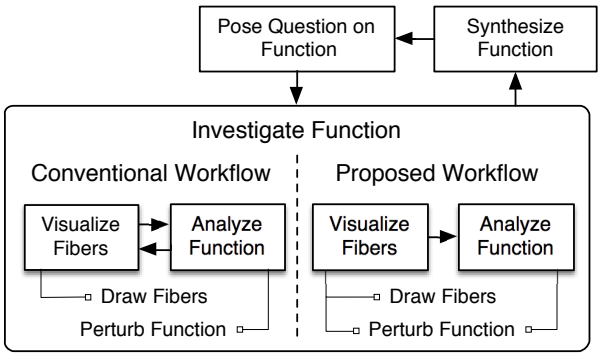


Fig. 6. A typical workflow for studying singular fibers using manual visualization. A mathematician poses a question on a function, then investigates the function by manual analysis (e.g. transforming equations) and sketching. The function is perturbed if degeneracies are found. The investigation continues until the function has been explored, and then a new function is selected, often by modifying the previous function. The process then iterates.

same as analyzing the Reeb space. Equally, Edelsbrunner et al. [12] 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 [14], and can now be seen to be identical to the Stein factorization [24]: in neither case was a practical computation given.

The Jacobi set has since been simplified [4, 33] and applied to ridge-valley extraction [25]. However, in these cases, the Jacobi set is illustrated in the $\boldsymbol{R}^2 \rightarrow \boldsymbol{R}^2$ 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 $\boldsymbol{R}^2 \rightarrow \boldsymbol{R}$ case, the Jacobi set is the set of critical points (extrema and saddles), but the critical contours are needed to identify features of uniform topological behavior. Correspondingly, in the $\boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$ case we are considering, the Jacobi set is a set of 1-manifold structures, as can be seen in Bremer et al. [5, Fig. 5(a)]. It then follows that the Reeb space, not just the Jacobi set, needs to be computed to support the mathematical workflow.

More recently, Carr and Duke approximated the Reeb space with the *Joint Contour Net* [6]. 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. This means that each pixel corresponds 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. In our approach, we employ it to extract the fiber topology of the function $\boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$ within the original finite 3D domain and separately for the restriction to the boundary.

Visually, Duke et al. [11] arranged JCNs aesthetically in 3D space for arbitrary dimension of the range space, and applied it to analyzing real datasets. In our case, however, it is better to keep the connection with the range, and this requires a new layout algorithm (Section 6).

4 Tasks for Studying Singular Fibers

With the mathematical preliminaries out of the way, we now turn to the question of how to support mathematical development of singular fiber theory. As with any application, this requires understanding the workflow into which the application will fit, as shown in Figure 6.

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 6, this starts with synthesizing a suitable function, and posing a question. Such a question can be how a degenerate singular fiber component can be topologically resolved to non-degenerate ones when the function is perturbed.

The mathematician then 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, 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 tasks and how machine-based visualization can help.

4.1 Drawing the Fibers

As shown in Figure 3, 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 [24, 14]. Although singular fiber theorists have considered the concept [19], 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 [6], we have added to the application the ability to visualize the Reeb space directly, as well as its relationship to the conventional range diagram.

4.2 Perturbing the Function

As noted above, singular fiber theory, like Morse theory, is the most straightforward if no degenerate fiber components exist, and this is usually handled by adding small perturbations to the function to destroy the degeneracies. Importantly, functions without degeneracies still approximate those with degeneracies, making perturbation a common strategy in studying fiber topology. In practice, perturbation tasks occur 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. Thus a considerable amount of trial and error is required in practice.

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.

5 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, 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 2D rendering of the fibers and the singular values. 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 1.

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

5.1 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 2, 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 1 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 zoomed and rotated, but this should primarily be understood as a passive view in which the results of choices made in the range view are displayed. For convenience, a pop-up menu is provided in this view so that the user can among other things perform screen captures of any view.

5.2 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, 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 5), 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 4, by providing inset windows showing the fibers in the domain view at a particular pixel, and by allowing the user to select such a pixel to be investigated. 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 4), 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, the 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, dark grey corresponds to one fiber component (i.e. a simple arc or a loop), and 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 now be seen as a projection of the graph of the function akin to that used in the work on continuous scatterplots [1].

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 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 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 6.3.

5.3 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 represented in the range view. As with the domain view, zooming and rotation is supported.

5.4 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. 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 6.5 and the topology of the function is recomputed and redrawn.

6 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. Note here that a fiber component is equivalent to a point in

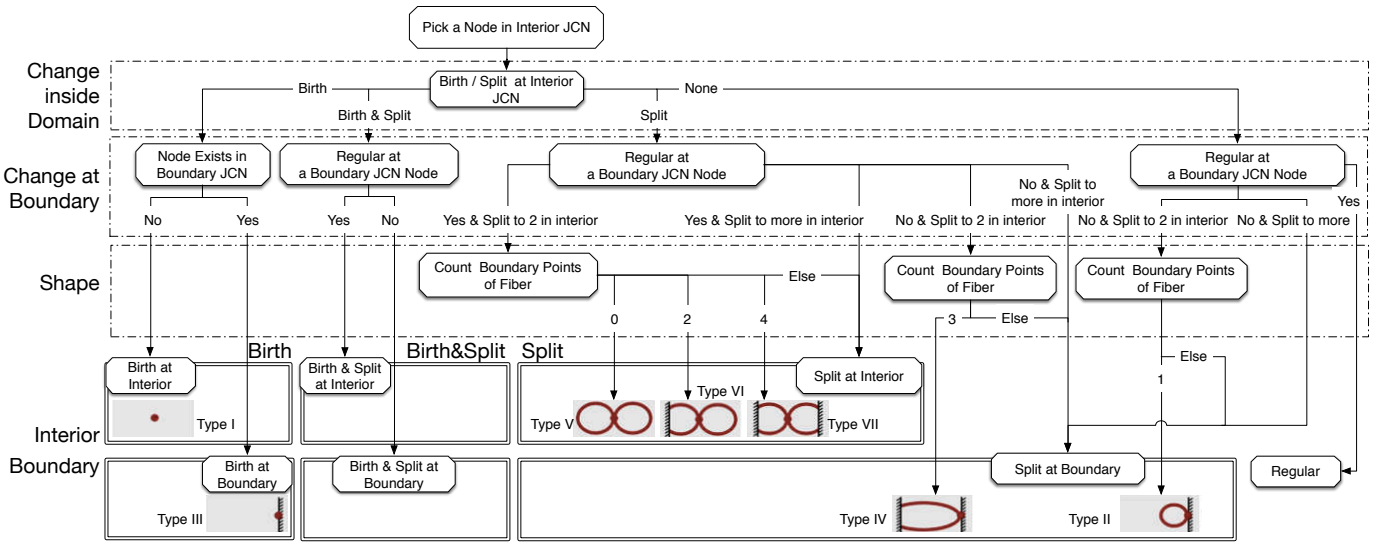


Fig. 7. Decision tree for identifying singular fibers in the JCN, based on Figure 4. We identify not only the non-degenerate singular fiber components (shown by icons), but also the degenerate ones (with double lines). Birth/split detection is illustrated in Figure 8.

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 [12].

Instead of this, we therefore approximate the topology of fibers by computing the Joint Contour Net (JCN) [6], as discussed in Section 6.1. 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 of nodes in the JCN, as described in Section 6.2.

Other tasks required include placement of the insets in the range view (Section 6.3), rendering the isosurfaces and fibers in the domain view (Section 6.4), editing the function in the domain view to remove degeneracies (Section 6.5), and laying the JCN out in the Reeb space view (Section 6.6).

6.1 Representing Fibers with the Joint Contour Net

Recall that the JCN approximates the Reeb space by quantizing the function values. Each node in it corresponds to a class of fibers defined by the Euclidean product of intervals in the range. In the $\boldsymbol{R}^3 \rightarrow \boldsymbol{R}^2$ case, these are rectangles in the range: i.e. pixels, and the JCN can be computed using rasterization algorithms [6]. An intuitive description with the connection to formal mathematics is found in literature [30].

In the present application, the function is sampled at a domain resolution of $50 \times 50 \times 50$, although this can easily be varied. The resulting cubic lattice is then subdivided into six tetrahedra per cell and rasterized to compute the JCN, using RGBA channels in the frame buffer to represent pixels in the range.

6.2 Identifying Singular Fibers

As seen in Section 2, the key task in singular fiber theory is to identify singular fibers and the topological changes that occur at them, as shown in Figure 4. 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 a regular fiber component will be adjacent to exactly four other nodes: one each above, below, to the left and to the right, in the $\boldsymbol{R}^2$ range. Any node that does not have exactly these four neighbors is therefore presumed to be singular.

Even further, the shape of each singular fiber component is deduced. Inspired by the classification in Figure 4, we refer to the changes in the numbers of fiber components and endpoints in the fiber. Notice that we can determine whether a fiber component (i.e. a node in the JCN) has a split/merge or birth/death, as shown in Figure 8. More precisely, a fiber component where a split/merge occurs will have at least two neighbors in one direction, while a fiber at which a new component appears/disappears will have no neighbors. 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). Following the original rasterization algorithm [6], we construct the interior JCN by rasterizing all of the triangles that constitute the overall tetrahedral tessellation. If two triangle fragments in the same pixel belongs to a common tetrahedron, they are grouped together to form a JCN node. In other words, each JCN 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, which is defined to be a set of tetrahedra/triangles that encloses a fiber component in the domain/boundary. Consider a node of the interior JCN and a set of tetrahedra that encloses the corresponding fiber. The topology of the fiber on the boundary can be obtained by extracting nodes from the boundary JCN in such a way that their associated triangles are contained in the aforementioned set of tetrahedra.

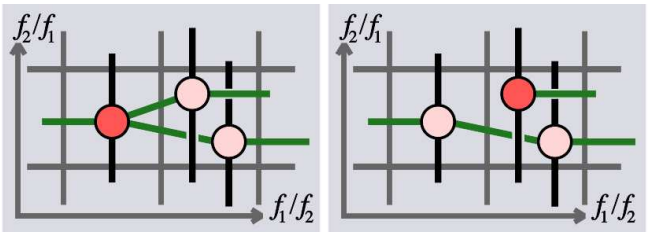
We iterate through the interior JCN nodes to determine the topological shape of each fiber component by consulting the decision tree in Figure 7. The idea of the tree is to classify fiber components to three categories: singular ones with/without singular points on boundary, and regular ones. The singular fiber components are further classified into "birth," "split," and their mixture. This allows us to classify any non-degenerate singular fiber component with a pattern in Figure 4.

More complex non-degenerate singular fibers can still be born or merged at multiple singular points. Discriminating these cases against those in Figure 4 was not achieved yet, as this requires investigating the number of singular points. This eventually leads to a separate work, since no singular point extraction algorithm is consistent with the approximation in the JCN. In practice, however, our classification algorithm satisfied the users as described later in Section 8.

6.3 Placing Inset Views

We saw in Section 5.2 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 [3], which arranges large annotation labels outside a central content area



(a) (b)

Fig. 8. Identifying topological change (in red) from the JCN. (a) Split/merge of a fiber component. (b) Birth/death of a fiber component.

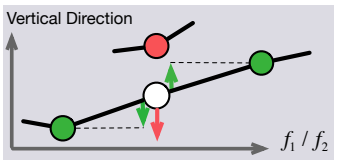


Fig. 9. For controlling the 3D layout of the interior JCN in the Reeb space view, each node in the JCN moves vertically along the z axis. The motion is controlled with the repulsive forces from other nodes having the same quantized (f_1, f_2) value (in red), and the attracting forces from the adjacent nodes (in green).

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 each inset with an optional black boundary as seen in Figures 1 and 10.

6.4 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 identified by extracting the level set of f_2 from the patches of marching tetrahedra for f_1.

6.5 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 topological 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 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 are in touch, the singular point will have its value changed and will instead occur at another values.

More specifically, 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 just need to locally displace one of the isosurfaces that contain the corresponding singular fiber.

Once the problem is stated in this way, the solution is straightforward. We allow the user to select any point on either isosurface and drag it inwards or outwards, for example, along the normal to the isosurface. Here, the change is accomplished by adding a small radial basis function (RBF), the amplitude of which is governed by the distance of the mouse pointer, and the radius of which is a small distance chosen as an initial parameter: half of the size of the domain by default in our implementation. Our experiment showed that the radius was large enough for avoiding too small perturbations to the topology.

6.6 3D Layout of the Reeb space

In the Reeb space view (Section 5.3), 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. Note here that we focus on the visualization of the interior JCN only because it clearly designates how the fiber evolves in terms of the two function values. Unfortunately, configurations exist in which we inevitably incur self-intersections of the Reeb space, making a canonical crossing-free algorithm impossible.

Since this exact problem arose in work on the contour tree, and required separate work to resolve [18], we can identify that this is a subtask beyond the scope of the present paper. Thus, like the solution used by Duke et al. [11], a heuristic layout is called for, based on a spring layout algorithm, shown in Figure 9. The popular force-directed graph layout algorithm [15] in 2D works fine for the 3D layout of the Reeb space in our experiments, but with a few modifications. We constrain (x, y) coordinates of each JCN node to be fixed at the two function values, while leaving the z coordinate available to the layout algorithm. Let us denote by a *sheet*, the cluster of nodes that belongs to the same connected component in a regular fiber. The z values of the JCN nodes are equalized if they constitute the same sheet, and spaced evenly along the z axis if multiple sheet exists at the corresponding (x, y) coordinates. Note that we set the z value of the other nodes to the mean of those of the adjacent nodes.

As illustrated in Figure 9, we apply to a node i the summation of attractive forces F_a exerted by its adjacent nodes, and repulsive forces F_r by nodes that share the same (x_i, y_i) coordinates as i. Here, $F_a(d_a) = {d_a}^2/k_i$ where d_a is the distance from an adjacent node, and $F_r(d_r) = -{k_i}^2/d_r$ where d_r is the distance from the node that shares (x_i, y_i). Moreover, control parameter k_i is empirically set as $k_i = C \times (\#\{\text{nodes at } (x_i, y_i)\})^{-1/2}$, and $C = \#\{\text{nodes in sheet}(i)\}/\max_j\{\#\{\text{nodes in sheet}(j)\}\}$, where sheet$(i)$ indicates the sheet containing i. Notice that each node moves along the z axis only and its displacement is bounded by some constant U that gradually decreases during the iteration to avoid any local minimum.

7 Implementation

We have implemented our system in C++ using Qt for the interface, CGAL for tessellating the domain space, Boost Graph Library for handling graphs, OpenGL for rendering mathematical objects and rasterizing the tessellated domain space for JCN construction.

Although performance is not the principal concern in this implementation, we give some typical figures for computation times. The computation for the sampling, rasterization, and JCN construction in total took 123 seconds for Figure 1, 255 seconds for Figure 10 (top), and 89 seconds for Figure 10 (middle). These times were recorded on a desktop PC with Intel Xeon E5 CPU with 6 cores (3.5 GHz, 256 KB L2 Cache per core and 12 MB L3 Cache), 32GB RAM, and an AMD Dual FirePro D500 GPU (3GB VRAM $\times$ 2).

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.

8 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 8.1, and by reporting feedback provided by some indicative users in Section 8.2.

It is the nature of this type of application that the initial target audience is small and highly specialized, and thus the participants of our user study include a research group headed by one of the co-authors.

Moreover, although tasks have been defined in Section 4, they are not readily fit to quantitative assessment, and it is more important to understand whether the target group feels that the application is successful in supporting their research. We therefore recruited 2 graduate students and 4 experienced researchers. Of the latter, 2 were co-authors, 1 a member in a co-author's group. We demonstrated the application to them, and collected their feedback when done.

8.1 Task Support

As a demonstration of how the application supports the desired tasks, professional mathematician, who participated outside of our group, used the interface to understand fibers in a particular function: $f(x,y,z) = (x^2, y^2+z^2)$. He employed this function as an example for understanding new results on fiber topology of boundaries [31]. We asked him to focus on the two principal tasks from Section 4: drawing the fibers and perturbing the function.

As shown in Figure 10, he firstly tracked the fiber topology of the function by dragging the cross-mark in the range view, which corresponds to the isosurfaces and fiber in the domain view. He then moved on to the second task, which was to perturb the function. His intention was to explore the possible deformations that resolve specific types of degeneracies, which leads to meaningful insights into many unsolved problems in mathematics. He began by using the traditional technique: perturbing the function with equation. He altered the input equation to $f(x,y,z) = (x^2, y^2+z^2+x)$, and then moved the cross-mark around in the range to understand the configurations of the fibers in the domain. After that, he interacted with the proposed RBF perturbation. In contrast to the traditional perturbation on equations, the RBF perturbation immediately showed how the fibers can be modified.

When drawing the fibers, he also placed inset views to remind himself of the topology. He reported that interactions in the domain view with other two views are significantly more intuitive, compared to traditional techniques.

The interface has let the user quickly understand the fiber topology of $f(x,y,z) = (x^2, y^2+z^2)$, and he found a few new topological deformations to the fibers. He reported that the visual indication provided was indeed useful since the task had been difficult to do manually. He also mentioned that the 3D Reeb space view was useful for understanding the connectivity of the fiber components and compensated for missing information in the range view. To demonstrate practical usages, a prototype version of our interface visually verified a hypothesis in mathematics [30, Fig. 17].

8.2 Observations

As in Section 8.1, we asked the participants to draw fibers and perturb functions. All participants agreed that the interface successfully provided an intuitive means of analyzing the singular fibers.

The graduate students commented that the domain was useful in elucidating cases where the fibers were knotted, as well as for understanding complicated merging/splitting of fibers. The experienced researchers felt that the rapid display of the overall configuration of the range was beneficial in understanding the global features of a function. The experienced researcher in Section 8.1 again reported that the Reeb space view helped understand the connectivity of the fiber components. When the Reeb space view contains multi-layered sheets with respect to the function values, however, the users often prefer the range view to identify the connectivity between fiber components.

Solving degeneracies also attracted considerable attention from the users. The graduate students reported that our interface allowed them to imagine various ways of solving a degeneracy. In contrast, the principal researcher supported the use of the interface for three reasons. First, it simplifies understanding the topological shape of a degenerate fiber by showing the decomposed non-degenerate singular fibers. Secondly, also as a classic visualization task, generating specific examples for illustrations becomes considerably easier. Finally, 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 users also felt that while the application was useful, they wanted us to enhance the usabilities. In particular, the users found that the interaction in the RBF perturbation should be more intuitive. Moreover, as a minor update, the users demanded the manual specification of function values in the range view, instead of mouse click in the window. Notwithstanding this, they felt that the application would reduce the difficulty of their study.

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

8.3 Limitations

Our visualization has been implemented at the cost of two major limitations. The first limitation arises from the Reeb space representation for visualizing the fiber topology and the second is caused by the manual perturbation.

The former poses a problem since the Reeb space inherently composes a multi-layered structure and cannot be embedded in a 2D space. This limits the complexity of input functions. Interactions between dozens of fiber components are impossible to be investigated with our interface. Especially, real world datasets have even more of them. Note that, however, this does not degrade the value of our work, because the examples in this paper are already challenging to investigate without our interface. On top of this, the JCN quantizes the function values, hiding the small deviations of the values. In other words, users cannot find small-scale topological transitions of fibers. On the other hand, it is still true that the approximate representation of Reeb space facilitates us to explore the global behavior of the targeted multi-fields, which is often preferable in the context of mathematical visualization.

As the second limitation, our manual perturbation interface gives no guarantee to decompose the specified degenerate singularity into non-degenerate ones and to keep a specific way of decomposing it. Nevertheless, this simultaneously provides enough degrees of freedom to investigate all the possibilities to resolve the degeneracy, especially for domain experts in the field. In practice, our users told us that such functionality also exhibits mathematical value, as in the scenario of Figure 10. Preparing interactions for more systematic perturbation remains to be tackled in collaboration with expert mathematicians.

9 Conclusion

This paper has presented a mathematical framework and a computational application for interactively visualizing fiber singularities of analytic multivariate functions $\boldsymbol{R}^3 \to \boldsymbol{R}^2$. This involved constructing JCNs for both the domain and its boundary, then using the connectivity of both to identify the types of singular fibers, and rendering the result in the range, domain, and Reeb space. Finally, the interface also supported online perturbation of the function to resolve degeneracies in the singular fibers.

In the future, we would like to design an easy-to-use interaction in RBF perturbation for users unfamiliar with 3D interactions, suppress artifacts arising from the approximate representations of Reeb space using JCNs. We also would like to let the user analyze the relationships between global and local topological properties. Further work on adaptive sampling of the JCN is under consideration, as is the development of perturbations that allow local updates to the JCN rather than global recomputation. To address the limitations described in Section 8.3, we would like to support various forms of analysis, including that for real world data. As an example, the approach may be extended for analyzing a time-varying scalar field by setting f_1 to the scalar value and f_2 to the time. In this case, the singular fiber components of birth/death correspond to local extrema and those of merge/split to saddles. Finally, it is our intention to build upon this interface to support analysis of higher-dimensional cases such as functions of the form $\boldsymbol{R}^4 \to \boldsymbol{R}^3$.

Acknowledgments

This work has been supported in part by EPSRC Grant EP/J013072/1, MEXT KAKENHI Grant 25120014, and JSPS KAKENHI Grants 23244008, 23654028, 25540041, 26730061, and 15K13438.

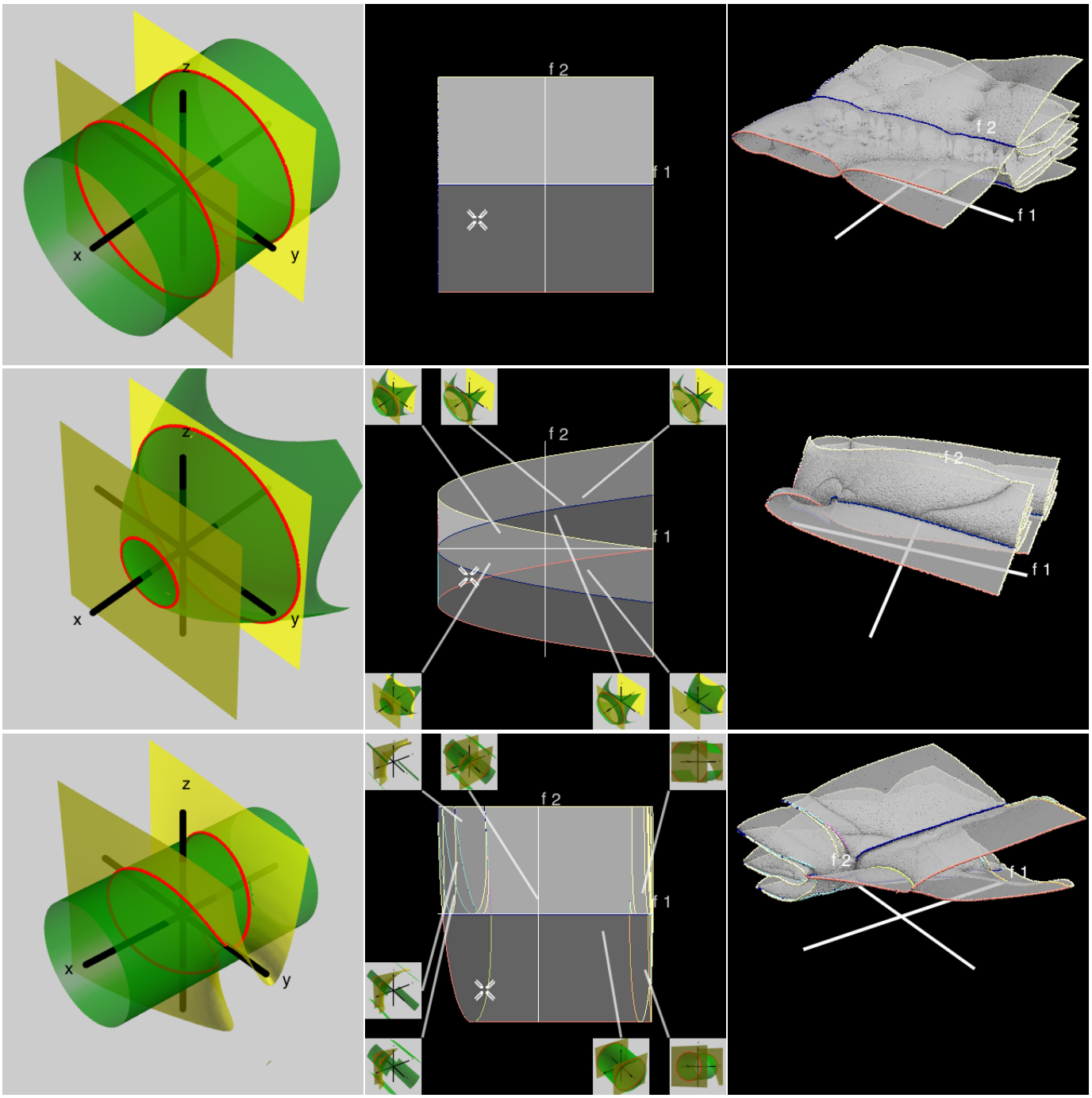


Fig. 10. The scenario described in Section 8.1. The expert used the interface to investigate how a perturbation can alter the given function, $f(x,y,z) = (x^2, y^2+z^2)$ (top), changed the equation to $f(x,y,z) = (x^2, y^2+z^2+x)$ to conduct the perturbation as the traditional technique in the field (middle), and ended up perturbing the function directly by deforming the isosurface with the RBF interface (bottom).

REFERENCES

[1] S. Bachthaler and D. Weiskopf. Continuous scatterplots. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1428–1435, 2008.

[2] C. L. Bajaj, V. Pascucci, and D. R. Schikore. The contour spectrum. In *Proceedings of IEEE Visualization '97*, pages 167–173, 1997.

[3] 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.

[4] 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.

[5] 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.

[6] H. Carr and D. Duke. Joint contour nets. *IEEE Transactions on Visualization and Computer Graphics*, 20(8):1100–1113, 2014.

[7] H. Carr, Z. Geng, J. Tierny, A. Chattopadhyay, and A. Knoll. Fiber surfaces: Generalizing isosurfaces to bivariate data. *Computer Graphics Forum*, 34(3):241–250, 2015.

[8] 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.

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

[10] K. Cole-McLaughlin, H. Edelsbrunner, J. Harer, V. Natarajan, and V. Pascucci. Loops in Reeb graphs of 2-manifolds. In *Proceedings of the Nineteenth Annual Symposium on Computational Geometry*, SCG '03, pages 344–350, New York, NY, USA, 2003. ACM.

[11] D. Duke, H. Carr, A. Knoll, N. Schunck, H. A. Nam, and A. Staszczak. Visualizing nuclear scission through a multifield extension of topological analysis. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2033–2040, 2012.

[12] H. Edelsbrunner and J. Harer. Jacobi sets of multiple Morse functions. In *Foundations of Computational Mathematics*, pages 37–57, 2004.

[13] H. Edelsbrunner, J. Harer, A. Mascarenhas, V. Pascucci, and J. Snoeyink. Time-varying reeb graphs for continuous spacetime data. *Computational Geometry*, 41(3):149 – 166, 2008.

[14] 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.

[15] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software: Practice and Experience*, 21(11):1129–1164, 1991.

[16] R. Fuchs and H. Hauser. Visualization of multi-variate scientific data. *Computer Graphics Forum*, 28(6):1670–1690, 2009.

[17] 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.

[18] 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.

[19] J. T. Hiratuka and O. Saeki. Triangulating Stein factorizations of generic maps and Euler characteristic formulas. *RIMS Kokyuroku Bessatsu*, B38:61–89, 2013.

[20] L. Huettenberger, C. Heine, H. Carr, G. Scheuermann, and C. Garth. Towards multifield scalar topology based on Pareto optimality. *Computer Graphics Forum*, 32(3pt3):341–350, 2013.

[21] L. Huettenberger, C. Heine, and C. Garth. Decomposition and simplification of multivariate data using Pareto sets. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2684–2693, 2014.

[22] 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.

[23] D. J. Lehmann and H. Theisel. Discontinuities in continuous scatter plots. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1291–1300, 2010.

[24] 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.

[25] G. Norgard and P.-T. Bremer. Ridge-valley graphs: Combinatorial ridge detection using Jacobi sets. *Computer Aided Geometric Design*, 30(6):597–608, 2013.

[26] 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.

[27] V. Pascucci and K. Cole-McLaughlin. Efficient computation of the topology of level sets. In *Proceedings of IEEE Visualization 2002*, pages 187–194. IEEE Computer Society Press, 2002.

[28] 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.

[29] O. Saeki. *Topology of Singular Fibers of Differentiable Maps*, volume 1854 of *Springer Lecture Notes in Mathematics*. 2004.

[30] 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.

[31] O. Saeki and T. Yamamoto. Singular fibers of stable maps of 3-manifolds with boundary into surfaces and their applications. *preprint*, 2014.

[32] 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.

[33] N. Suthambhara and V. Natarajan. Simplification of Jacobi sets. In V. Pascucci, X. Tricoche, H. Hagen, and J. Tierny, editors, *Topological Methods in Data Analysis and Visualization: Theory, Algorithms, and Applications*, pages 91–102. Springer-Verlag Berlin Heidelberg, 2011.

[34] 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.

[35] S. Takahashi, Y. Takeshima, and I. Fujishiro. Topological volume skeletonization and its application to transfer function design. *Graphical Models*, 66(1):24–49, 2004.

[36] 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.