Innovative Approaches of Data Visualization and Visual Analytics

Chapter 4: Feature-Based Uncertainty Visualization

Keqin Wu,
University of Maryland Baltimore County
USA

Song Zhang,
Mississippi State University
USA

ABSTRACT

While uncertainty in scientific data attracts an increasing research interest in the visualization community, two critical issues remain insufficiently studied: (1) visualizing the impact of the uncertainty of a data set on its features and (2) interactively exploring 3D or large 2D data sets with uncertainties. In this chapter, a suite of feature-based techniques is developed to address these issues. First, an interactive visualization tool for exploring scalar data with data-level, contour-level, and topology-level uncertainties is developed. Second, a framework of visualizing feature-level uncertainty is proposed to study the uncertain feature deviations in both scalar and vector data sets. With quantified representation and interactive capability, the proposed feature-based visualizations provide new insights into the uncertainties of both data and their features which otherwise would remain unknown with the visualization of only data uncertainties.

1 INTRODUCTION

Uncertainty is a common and crucial issue in scientific data. The goal of uncertainty visualization is to provide users with visualizations that incorporate uncertainty information to aid data analysis and decision making. However, it is challenging to quantify uncertainties appropriately and to visualize uncertainties effectively without affecting the visualization effect of the underlying data information.

Uncertainty in scientific data can be broadly defined as statistical variations, spread, errors, differences, minimum maximum range values, etc. (Pang, Wittenbrink, & Lodha, 1997) This broad definition covers most, if not all, of the possible types and sources of uncertainty related to numerical values of the data. In this chapter, however, we investigate the uncertain positional deviations of the features such as extrema, sinks, sources, contours, and contour trees in the data. These feature-related uncertainties are referred to as feature-level uncertainty while the uncertainties related to the uncertain numerical values of the data are referred to as data-level uncertainty. Visualizing feature-level uncertainty reveals the potentially significant impacts of the data-level uncertainty, which in turn helps people gain new insights into data-level uncertainty itself. Therefore, investigating the uncertainty information on both data-level and feature-level provides users a more comprehensive view of the uncertainties in their data.

Many uncertainty visualizations encode data-level uncertainty information into different graphics primitives such as color, glyph, and texture, which are attached to surfaces or embedded in volumes (Brodlie, Osorio, & Lopes, 2012; Pang et al., 1997). Those methods. In essence, give global insight into the data by differentiating the area of high uncertainty from that of low uncertainty, however, the impact of the uncertainty on the important features of the data is hard to assess in such visualizations. Meanwhile, uncertainty visualizations may be subject to cluttered display, occlusion, or information overload due to the large amount of information and interference between the data and its uncertainty. We believe that one promising direction to cope with this challenge is to allow users to explore data interactively and to provide informative clues about where to look.

In this chapter, while our uncertainty representation can be adapted to different uncertainty models, we measure the uncertainty according to the differences between the data values, critical points, contours, or contour trees of different ensemble members. Our objectives are (1) to bring awareness to the existence of feature-level uncertainties, (2) to suggest metrics for measuring feature-level uncertainty, and (3) to design an interactive tool for exploring 3D and large 2D data sets with uncertainty.

In what follows, we first discuss related work, issues, and challenges of uncertainty visualization in section 2, then explain in detail in section 3 and 4 our methods: (1) an interactive contour tree-based visualization for exploratory visualization of 2D and 3D scalar data with uncertainty information and (2) a framework for visualizing feature-level uncertainty based on feature tracking in both scalar and vector fields. Lastly, we conclude this chapter and discuss future directions.

2 BACKGROUND

We discuss issues, challenges, and the related work of uncertainty visualization in this section.

2.1 The Gap between Data-Level Uncertainty and Feature-Level Uncertainty

Knowing the uncertainty concerning features is important for decision making. Many uncertainty visualizations based on statistical metrics merely measure uncertainty on the data-level—the uncertainty concerning the numerical values of the data and introduced in data acquisition and processing. While these techniques achieved decent visualization results, they do not provide users enough insight into how much uncertainty exists for the features in the data. For example, the uncertainty of the ocean temperature data may result in the uncertain deviation of the center of an important warm eddy. The uncertainty of the hurricane wind data may cause the uncertain location of a hurricane eye. This kind of uncertainty is neglected by many current methods but needs to be quantified and visualized so viewers are aware of it.

In scientific data, the difference between a known correct datum and an estimate is among the uncertainties most frequently investigated. To compare data-level uncertainty and feature-level uncertainty, we investigate two data sets. The first data set is a slice of a simulated hurricane Lili wind field (Figure 1a). The second data set is created by adding random noise to the first dataset. For a wind vector, its data-level uncertainty is represented as both angular difference and magnitude difference between vectors of the two fields. Figure 1c shows the arrow glyphs (Wittenbrink, Pang, & Lodha, 1996) for visualizing the uncertainty of vector fields with the angular uncertainty presented as the span of each arrow glyph and magnitude uncertainty as the two winglets around an arrow head. For details about designing arrow glyphs for vector field uncertainty, please refer to (Wittenbrink et al., 1996).

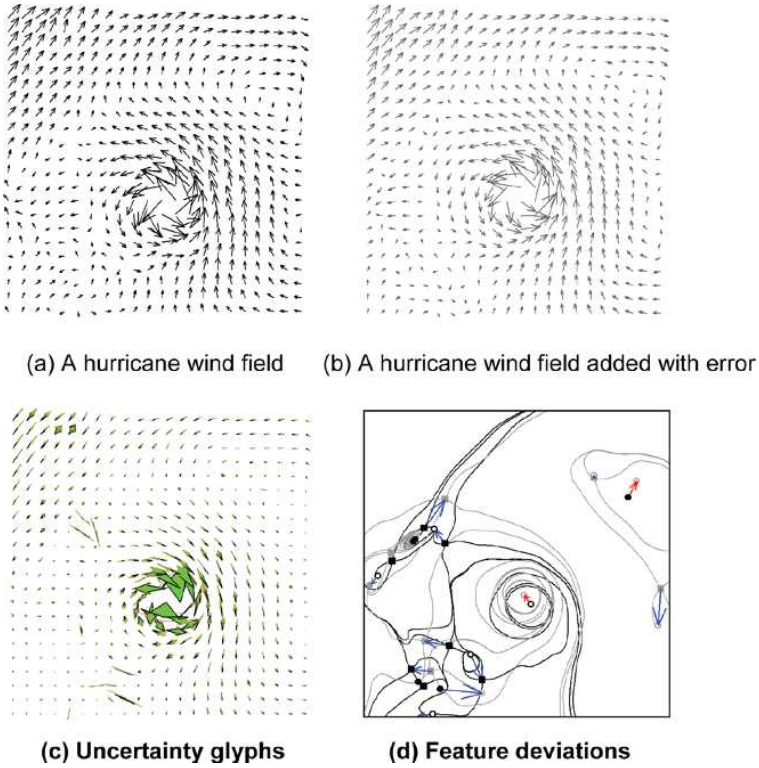

Figure 1: An uncertain vector field and its uncertain feature locations

As shown in Figure 1c, with the uncertainty glyphs, users may notice that the area around the hurricane eye (the major vertex in the middle) exhibits high data-level uncertainty which raises a question: does it affect the location of hurricane eye? Only an explicit comparison between the hurricane eyes in the two fields will answer this question. We therefore extract topologies of the two fields as shown in Figure 1d. The sink point (in black) inside the hurricane eye noticeably shifts northwest (indicated by a red arrow) from the original vector field to the new one (in gray). Another question is, is there anything hidden in the relatively low uncertainty area indicated by the small arrow glyphs? A quick look at the Figure 1d reveals that the upper corner vortex significantly shifts its position (indicated by a red arrow) though it is located at the region with relatively low data-level uncertainty.

This example illustrates that merely investigating the data-level uncertainty may not tell the whole story about the data and that the feature-level uncertainty is an indispensable part of uncertainty that cannot be neglected. Moreover, visualizing uncertainty of features, instead of that of the data, may provide a succinct and meaningful representation of the uncertainty and thus give a better interpretation of the data.

2.2 Challenges of Visualizing Uncertainty

Representing uncertainty in 3D or large 2D data sets could encounter severe issues such as cluttered displays, information overload, and occlusions. Many uncertainty visualizations place glyphs that encode uncertainty within the visualization of the data. For instance, Sanyal et al. (2010) visualized data-level uncertainties via circular or ribbon-like glyphs over a color-mapped image of the data. Despite their effectiveness in revealing uncertainty information accurately at glyph locations, due to the overlaps between the data and uncertainty glyphs, the number of the glyphs has to be limited and information loss for both the data and uncertainty is unavoidable. Other techniques which overlay or embed uncertainty representation in the data visualization face similar issues.

Meanwhile, interaction with the visualization of 3D or large 2D data sets could encounter issues such as geometry bandwidth bottleneck, depth perception, occlusion, and inefficiency in 3D object selection. These issues are inherent in interactive visualizations, and may be intensified in the integrated visualization of a 3D dataset or large 2D data set and its uncertainty because there is simply more information to show in such visualizations.

2.3 Feature-Based Uncertainty Visualization

In feature-based visualization, features are abstracted from the original data and can be visualized efficiently and independently of the data. We believe that the feature-based uncertainty visualization becomes desirable when the size and complexity of the uncertainty information

increase.

Several features interest us the most. Critical points such as sinks, sources, maxima, and minima are representative of topological features that carry significant physical meaning of a scalar or vector data set (Helman & Hesselink, 1989; Keqin, Zhanping, Song, & Moorhead, 2010). Contours, including 2D iso-lines and 3D iso-surfaces, are features frequently investigated for exploring data with uncertainty (Grigoryan & Rheingans, 2004; Sanyal et al., 2010). A contour tree stores the nesting relationships of the contours of a scalar field. It is a popular visualization tool for revealing the topology of contours (Hamish Carr, Snoeyink, & Axen, 2003), generating seed set for accelerated contour extraction (Hamish Carr & Snoeyink, 2003), and providing users an interface to select individual contours (Bajaj, Pascucci, & Schikore, 1997).

Several methods have been developed to address the uncertainty of the size, position, and shape of contours (Grigoryan & Rheingans, 2004; Pfaffelmoser, Reitinger, & Westermann, 2011; Rhodes, Laramee, Bergeron, & Sparr, 2003). Rendering contours from all ensembles in a single image, known as spaghetti plots (Diggle, Heagerty, Liang, & Zeger, 2002), is a conventional technique used by meteorologists for observing uncertainty in their simulations. Pang et al. (Pang et al., 1997) presented fat surfaces that use two surfaces to enclose the volume in which the true but unknown surface lies. Pauly et al. (Pauly, Mitra, & Guibas, 2004) quantified and visualized the uncertainty introduced in the reconstructions of surfaces from point cloud data. Pfaffelmoser et al. (Pfaffelmoser et al., 2011) presented a method for visualizing the positional variability around a mean ISO-surface using direct volume rendering. A method to compute and visualize the positional uncertainty of contours in uncertain input data has been suggested by Pöthkow and Hege (Pöthkow & Hege, 2011). Assuming certain probability density functions, they modeled a discretely sampled uncertain scalar field by a discrete random field.

Among uncertainty methods, only a few are proposed to address the topological features. Otto et al. studied the uncertain topological segmentation of a vector field by introducing the probability density that a particle starting from a given location will converge to a considered source or sink (Otto, Germer, Hege, & Theisel, 2010). The uncertainty related to the topology structure of a scalar field, is barely studied.

2.4 Visual Encoding of Uncertainty

Several efforts have been made to identify potential visual attributes for uncertainty visualization. MacEachren suggested the use of hue, saturation, and intensity for representing uncertainty on maps (1992). Hengl and Toomanian (2006) showed how color mixing and pixel mixing can be used to visualize uncertainty in soil science applications. Davis and Keller (1997) suggested value, color, and texture for representing uncertainty on static maps. Djurcilov, Kima, Lermusiaxb, and Pang (2002) used opacity deviations and noise effects to provide qualitative measures for the uncertainty in volume rendering. Sanyal, Zhang, Bhattacharya, Amburn, and Morrehead (2009) conducted a user study to compare the effectiveness of four uncertainty representations: traditional error bars, scaled size of glyphs, color-mapping on glyphs, and color-mapping of uncertainty on the data surface. In their experiments, scaled sphere and color mapped sphere perform better than traditional error bars and color-mapped surfaces. Later, they proposed graduated glyphs and ribbons to encode uncertainty information of weather simulations (Sanyal et al., 2010).

While the uncertainty visualization is application-dependent in many cases, two visualization schemes are widely used: using intuitive metaphors, such as blurry and fuzzy effects (Cedilnik & Rheingans, 2000; Djurcilova et al., 2002; Grigoryan & Rheingans, 2004), which naturally implies the existence of uncertainty, and using quantitative glyphs (Sanyal et al., 2010; Schmidt et al., 2004), which shows quantified uncertainty information explicitly. Both schemes have their own tradeoffs. In quantitative glyphs the uncertainty information has to be shown in a discrete way. By using uncertainty metaphors, people get less quantified information about uncertainty since they cannot tell levels of blur or fuzziness apart accurately (Kosara et al., 2002). To reveal uncertainty accurately, uncertainty glyph is preferred to uncertainty metaphors.

3 AN INTERACTIVE FEATURE-BASED VISUALIZATION OF SCALAR FIELD UNCERTAINTY

In this section, we analyze uncertainty-related information on three levels: on data-level, we study the uncertainty of the data, on contour-level, we quantify the positional variation of the contours, and on topology-level, we reveal the variability of the contour trees.

The core of this method is the use of contour trees as a tool to represent uncertainty and to select contours accordingly. First, a planar contour tree layout which suppresses the branch crossing and integrates with tree view interaction is developed for a flexible navigation between levels of detail for contours of 3D or large 2D data sets. Second, we attach uncertainty information to the planar layout of a simplified contour tree to avoid the visual cluttering and occlusion of viewing uncertainty within volume data or complicated 2D data. We call the scalar field obtained by averaging the values from all the ensemble members at each data point the ensemble mean and the contour tree of the ensemble mean the mean contour tree. We show the data-level uncertainty by displaying the difference between each ensemble member and the ensemble mean at each data point or along a contour. For contour-level uncertainty, given a contour in the ensemble mean, we compute the mean and variance of the differences between this contour in the mean field and its corresponding contours in all the ensemble members. For topology-level uncertainty, we map the contour trees of all the ensemble members to the mean contour tree and show their discrepancy to indicate uncertainty.

This section is structured as follows. Section 3.1 provides the definition, simplification, layout, and tree view graph design of contour trees. The visualization of three levels of uncertainties is discussed in section 3.2. Section 3.3 introduces the interface design. Section 3.4 demonstrates and discusses application results.

3.1 Contour Tree Layout and Tree View Graph Design

We visualize uncertainty and variability information through contour trees. The layout of a contour tree becomes an issue when the size and complexity of the contour tree increase (Heine, Schneider, Carr, & Scheuermann, 2011). To explore data with uncertainty efficiently, a preferred contour tree layout is the one that is two-dimensional, shows hierarchy and height information intuitively, suppresses branch self-intersections, allows for a fast navigation through different levels of simplification, and allows displaying uncertainty information. To meet these requirements, we propose a planar contour tree layout which is integrated with the tree view graph interaction.

3.1.1 Contour Tree

The contour tree is a loop-free Reeb graph (Tierny, Gyulassy, Simon, & Pascucci, 2009) that tracks the evolution of contours. Each leaf node is a minimum or maximum, each interior node is a saddle, each edge represents a set of adjacent contours with iso-values between the values of its two ends. There is a one-to-one mapping from a point in the contour tree (at a node or in an edge) to a contour of the scalar field. A contour tree example is shown in Figure 2. More detailed information about contour and contour tree can be found in (Carr, 2004).

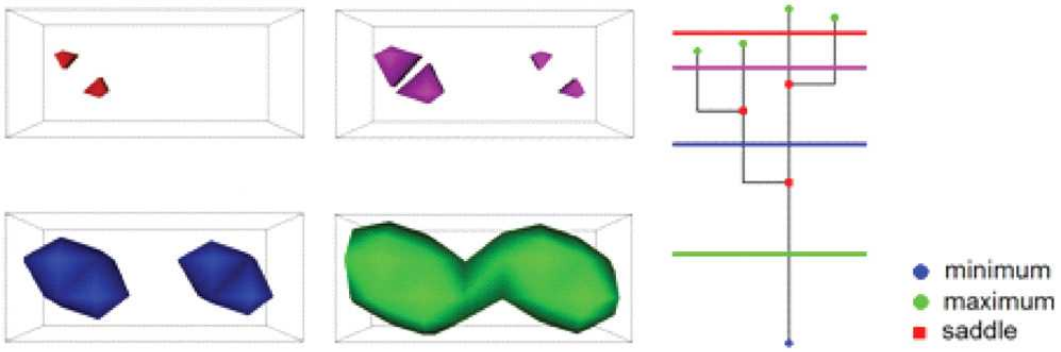

Figure 2: The contour tree and contours of a 3D scalar field Each horizontal line cuts exactly an edge of the tree for every contour at the corresponding iso-value Color is used to indicate the correspondence between a line and its corresponding contours

3.1.2 Contour Tree Simplification

Simplification is introduced to deal with the contour trees that are too large or complicated to be studied or displayed directly (Hamish Carr, Snoeyink, & Panne, 2004; Pascucci, Cole-McLaughlin, & Scorzelli, 2004). Contour tree simplification facilitates a high level overview of a scalar field. Particularly, a simplified contour tree attached with uncertainty glyphs reduces the workload in viewing 3D data or complicated 2D data with uncertainty.

Usually, a contour tree simplification is conducted by successively removing branches that have a leaf node (extremum) and an inner node (saddle) (Hamish Carr et al., 2004; Takahashi, Takeshima, & Fujishiro, 2004). Figure 3 shows a top-down simplification (Wu & Zhang, 2013) by repeatedly assembling the longest branches on a current contour tree. The black numbers 0, 1,…, 4, indicate the order of assembling the branches. As shown in this example, the order of assembling the branches suggests a balanced hierarchy. The shorter branches are found at lower hierarchies and higher branches are located at higher hierarchies so that the more simplified contour tree always catches the more significant features. Each branch rooted at an interior node, other than the two ends of the branch, is a child branch. A branch which has child branches is called the parent branch of its child branches. The sub-tree which consists of all the edges and nodes along a branch and its descendants is called the sub-tree of the branch.

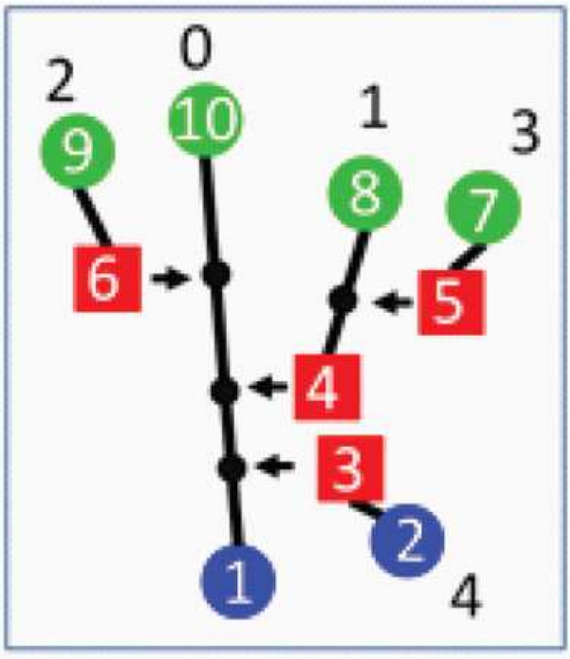

Figure 3: Contour tree hierarchy built based on a top-down contour tree simplification

3.1.3 A Rectangular Contour Tree Layout

The key to prevent unnecessary self-intersections is to recursively assigning a vertical slot to a branch so that its child branches are contained entirely within the slot.

To be more specific, a branch *B* is assigned a vertical slot *R*, and each child branch b_i of *B* is assigned a disjoint portion of *R*—a smaller vertical slot R_i within *R*. Nodes are positioned on the *y*-axis according to their function values. To emphasize the hierarchy, a branch is rendered in the middle, and its child branches are spread out to its left and right sides. The longest branch is drawn as a vertical line segment in the middle of the display. All the other branches have L shapes that connect extremum to their paired saddles to prevent them from crossing the slots of their siblings. An example is shown in Figure 4d and e. The rectangular display of the same tree reduces the number of crossings from three to one since the layout design rules out the case where a child branch intersects with its parent or siblings. Some self-intersections are unavoidable due to the strangulation cases where a downward branch appears as a child of an upward branch, or vice versa. A space-saving solution is given in Figure 4c which takes less horizontal space than the tree layout in Figure 4b. For a given parent branch, we separated it into three parts vertically: from high to low, upward branch zone where all the child branches are upward, mixed branch zone where child branches are either upward or downward, downward branch zone where all the child branches are downward. In the mixed branch zone, the upward branches take one side while the downward branch takes the other. In the upward branch or downward branch zone, the child branches stretch outward from the parent branch on both left and right sides without overlapping each other.

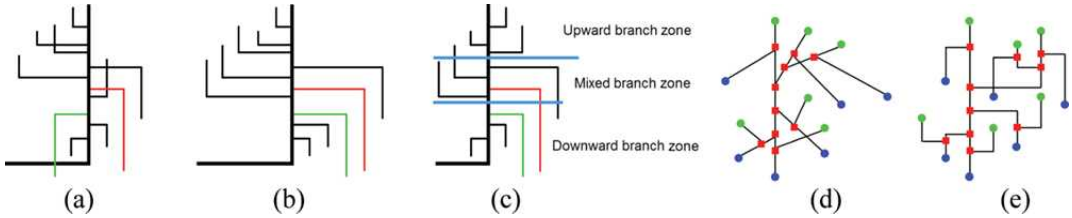

Figure 4: Strategies to reduce self-intersections (a) An unguided placement of child branches with multiple branch crossings (b) Placing upward child branches and downward child branches on the different sides (c) Placing upward child branches and downward child branches on the different sides only in the mixed branch zone (d) A 2D layout of a contour tree with 3 branch crossing (e) The rectangular display of the contour tree in d

3.1.4 Tree View Interaction Design

A typical tree view graph displays a hierarchy of items by indenting child items beneath their parents. In treeview representations, the interactions are directly embedded: the user can collapse (or expand) a particular sub-tree to hide (or show) the items in it.

An example of data exploration through tree view interaction is given in Figure 5. The data is a vorticity magnitude field of a simulated flow with vortices. A click on an inner node of the original contour tree (Figure 5a right) results in hiding or showing the sub-tree rooted at the node. The persistence – indicated as the vertical length of a branch –serves as an importance indicator for users to select contours. An interactively simplified contour tree is shown in Figure 5b. The roots of the collapsed sub-trees are marked with plus mark icons. For each branch of a contour tree, a single contour of the branch is extracted. A contour in the scalar field is represented by a point in the tree (indicated by a horizontal line segment in the same color with the contour). The selected contours are representative contours that give an overview of the whole scalar field. The more a contour tree is simplified, the higher level overview is obtained.

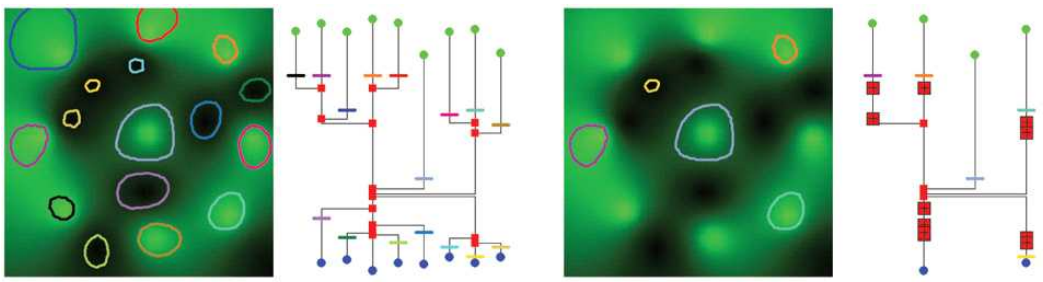

Figure 5: Tree view Interaction (a) A contour tree and a 2D scalar field from which this tree was constructed with contours selected for every branch of it (b) An interactively simplified contour tree after clicking on several nodes and contours selected for every branch of it

3.2 Contour Tree-Based Uncertainty Visualization

This section discusses uncertainty metrics for the data-level uncertainty and contour-level uncertainty and their visualization representations. The uncertainty or variability information is attached to the new planar contour tree display to give a high-level overview of uncertainty and to allow a quick and accurate selection of contours with different levels of uncertainty.

3.2.1 Data-Level Uncertainty

The data-level uncertainty measures how uncertain the numerical values of the data are. Uncertainty measures, such as standard deviation,

inter-quartile range, and the confidence intervals, fall into this category. As discussed in section 2.2, techniques which overlay or embed uncertainty representation in the underlining data visualization face perception and interaction issues. We therefore propose an alternative visualization that attaches uncertainty glyphs to a contour tree rather than integrating them with data visualization directly.

We adapted graduated glyphs (Sanyal et al., 2010) to visualize data-level uncertainty. A circular glyph encodes the deviation of all ensemble members from the ensemble mean at a data point. A graduated ribbon is constructed by interpolating between circular glyphs placed along an iso-line in the data image. A glyph that has a dense core with a faint periphery indicates that ensemble members have a few outliers and mostly agree. A mostly dark glyph indicates that large differences exist among individual members. The size of a glyph indicates the variability of a location with respect to other locations on the grid. For more details on graduated glyphs, we refer the reader to (Sanyal et al., 2010).

We use graduated glyphs to show uncertainty at each data point. Figure 6 shows a 2D 9×9 uncertain scalar dataset which is a down-sampled sub-region of the data in Figure 5. In Figure 6a, the mean field of eight members is color-mapped and overlaid with uncertainty glyphs at each data point. Likewise, as shown in Figure 6b, the glyph of a data point is attached to its corresponding location in the contour tree. Figure 6c illustrates a segment of a graduated ribbon along a branch. As shown in Figure 6b and d, while a graduated circle illustrates the data-level uncertainty at a data point, the graduated ribbon provides continuous uncertainty representation along individual contours with less clutter. Therefore, we prefer ribbon-like graduated glyphs over circular graduated glyphs for representing data-level uncertainty along contours.

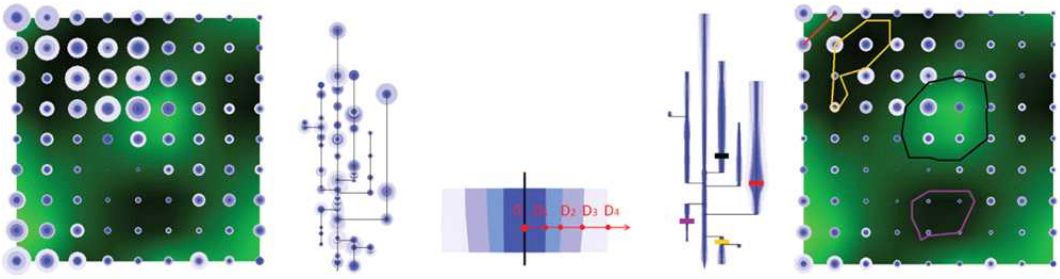

Figure 6: Data-level uncertainty representation based on the contour tree (a) Graduated circular uncertainty glyphs on an uncertain scalar field (b) A contour tree with circular graduated uncertainty glyphs (c) A segment of a graduated ribbon on a branch (black) constructed by overlaying thinner ribbons successively (d) Contour tree attached with average data-level uncertainty along each contour and four sets of corresponding contours are shown after clicking on four locations (indicated by colored line segments) in the contour tree

3.2.2 Contour-Level Uncertainty

In this chapter, contour-level uncertainty measures the variation in the position of a contour. In a spaghetti plot, the most unstable contours and the places where the contours are extremely diverse among individual ensemble members are interesting to users (Sanyal et al., 2010). However, the users' estimates tend to be inaccurate due to the randomness of the contour size, shape and length. Additionally, it is hard to look into such uncertainty in a large data set. Precise and automatic contour variability measurement is needed to assist the exploration of the large uncertain data. We introduce the concept of contour variability which quantifies how diverse a contour is within multiple ensemble members to address this need.

To measure the variability of a contour, we first identify corresponding contours in different ensemble members. Then, we calculate the differences between them and use the mean and variance of the differences to represent the positional variation among contours.

3.2.2.1 Contour Correspondence and Difference

For a contour in one ensemble member, there may be more than one contour in another ensemble member with the same iso-value with it, or it may be missing in some ensemble members. Spatial overlap is frequently used as a similarity measure to match features in different datasets under an assumption that these features only have small spatially deviation (Schneider, Wiebel, Carr, Hlawitschka, & Scheuermann, 2008; Sohn & Chandrajit, 2006). For instance, the correspondence between two contours can be measured by the degree of contour overlap as discussed by Sohn and Bajaj (Sohn & Chandrajit, 2006) when they computed correspondence information of contours in time-varying scalar fields.

The non-overlapped area $A(C,C_i)$ between the two corresponding contours *C* and C_i decides the difference between the two contours.

Given a contour *C* in the ensemble mean, we search in ensemble member *i* for the contour $C_i(i=1,\ldots,k)$ who shares the same iso-value with *C* and has the best correspondence with *C*. The best matched contours, if found, are considered the same contours which appear in different ensemble members. Figure 7 gives an example. In Figure 7a, there are three contours (in blue, gray, and brown) in ensemble *i* with different correspondence degrees with contour *C* (in red). With the largest overlap degree with *C*, the blue contour is identified as the corresponding contour of *C* in ensemble member *i*. Figure 7b, c and d illustrate the non-overlapping area in different cases. To reduce bias towards long or short contours, larger or smaller iso-surfaces, we normalize the non-overlapping area with the contour length (or iso-surface area in 3D case) of *C*: $difference(C,C_i) = A(C,C_i) / size(C)$ where *size(C)* is the contour length (or iso-surface area in 3D case) of *C*.

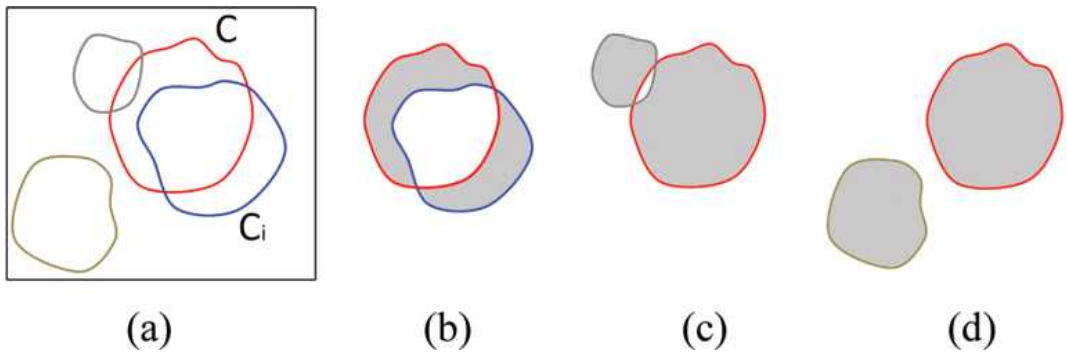

Figure 7: Contour difference measured by non-overlapped areas (a) Three contours in ensemble member *i* share a domain with contour *C* (in red) in the ensemble mean (b), (c), and (d) Non-overlapped areas (filled with gray) of different contours

3.2.2.2 Contour-Level Uncertainty Metrics and Visualization

We calculate the mean and variance of the differences to the mean to help a user select contours according to the quantified contour variability information. Given a contour *C* in the ensemble mean, let its corresponding contours of *k* individual ensemble members be $C_i (i = 1, \ldots, k)$ C_i is the contour with the same iso-value that is matched with *C*. The average difference among the corresponding contours is: $mean = \frac{\sum difference\left(C, C_i\right)}{k}$ The variance of difference among contours is: $va = \frac{\sum (difference\left(C, C_i\right) - mean)^2}{k-1}$ If an ensemble member does not have a matched contour for *C*, its contribution is set to a large value, the maximum non-overlapping area found between *C* and all the matched contours C_i

The contour variability along a contour can be shown at the corresponding location in the contour tree. Figure 8 illustrates the glyph design to encode the contour variability. Before visualization, both variability statistics are normalized to a range between 0 and a unit width. As discussed in section 3.2.1, ribbon-like glyph is preferred over circular glyph to prevent visual cluttering, we resample the varying contour variability along each branch and use linear interpolation to produce a ribbon-like glyph along each branch. For each branch, two ribbons are attached. The blue one is for the mean difference, while the green one is for the variance. The varying width of each ribbon indicates the varying magnitude of each variability measurement for the contours along the branch.

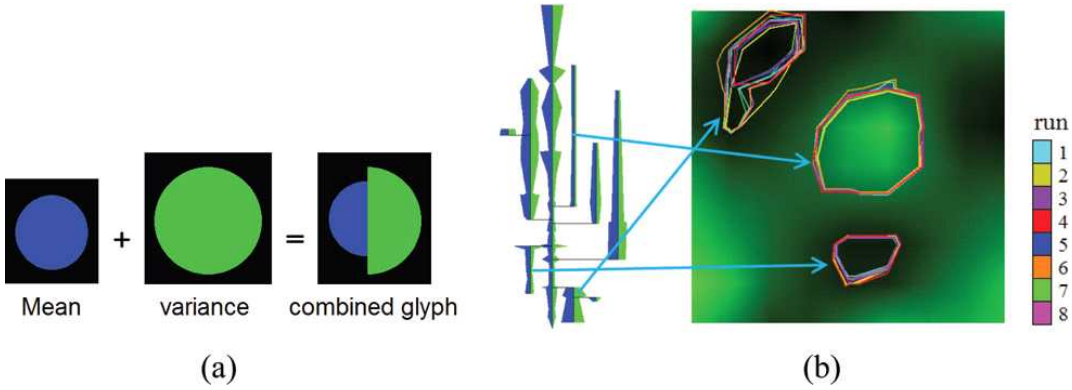

Figure 8: Visualization of contour variability based on contour tree (a) A circular glyph for contour variability is produced by combining two circular glyphs for mean and variance of differences between corresponding contours in different ensemble members and the ensemble mean (b) The ribbons attached to the contour tree (left) indicate the variability of corresponding contours in the data (right) Three sets of corresponding contours are shown after clicking on three locations (indicated by arrows) in the contour tree

3.2.3 Topology-Level Uncertainty

The uncertainty within the data impact not only the values or the contour positions locally, but also the global pattern of the data which is described by the topology of the data. Visualizing the uncertainty concerning the topology among the ensemble members provides new perspective on the global impact of uncertainty. In this chapter, the topology-level uncertainty is defined as the variation in the height and number of branches in the contour tree of an uncertain scalar field.

The idea is to map the branches between the contour tree of different ensemble members and the contour tree of the ensemble mean and to overlay the mapped branches. A set of matched branches are assigned with a same *x*-axis value but keep their original y-axis values so that an overlap of the branches on x-axis indicates their correspondence while the disagreements between the branches on y-axis indicate their discrepancy in iso-value. A branch of mean contour tree may not find a matched branch in some ensemble members. The number of matched branches is encoded with the width of the branch. A thicker branch is more certain than a thinner one.

3.2.3.1 Branch Correspondence

We measure the correspondence degree between two branches as the spatial overlap between their contour regions. A contour region of a branch is defined as the region covered by all the contours within the sub-tree of the branch.

Given a branch *B* in the mean contour tree, we search in ensemble member *i* for its best matched branch. The best matched branches, if found, are considered the same branches which appear in different ensemble members. We do not need to search all the branches of the contour tree in ensemble member *i* for the branch with largest overlap with *B*. The contour tree hierarchy and branch orientation help us limit the number of branches to compare. (1) The matched branch must be found among the child branches of the matched branch of its parent due to the nesting relationship between child and parent branches. (2) A downward branch does not match an upward branch, or vice versa.

Figure 9 illustrates the branch correspondences detected according to the spatial overlaps of contour regions. The contours and contour trees of two scalar fields in Figure 9a and b share a same spatial domain. In Figure 9a, the sub-tree of the middle branch (1, 10) is the whole contour tree. The sub-tree of branch (0.5, 10.5) is the whole contour tree in Figure 9b. Both branches share the whole data domain and hence are the best matched branches of each other. For the rest branches, the correspondence indicated by overlaps between their contour regions: branch (2, 3) → branch (1.8, 3.3), branch (6, 9) → branch (6.5, 9.6), branch (4, 8) →branch(4.2, 8.3), and branch(5, 7) → branch(5.4, 7.1). For example, branch (4, 8) matches branch (4.2, 8.3) since they have the largest overlaps. Branch (5.4, 7.1) matches branch (5, 7) even when branch (4, 8) has larger overlap with it since branch (4, 8) is mapped to its parent branch (4.2, 8.3). Figure 9c shows the matched branch in the same *x*-axis location.

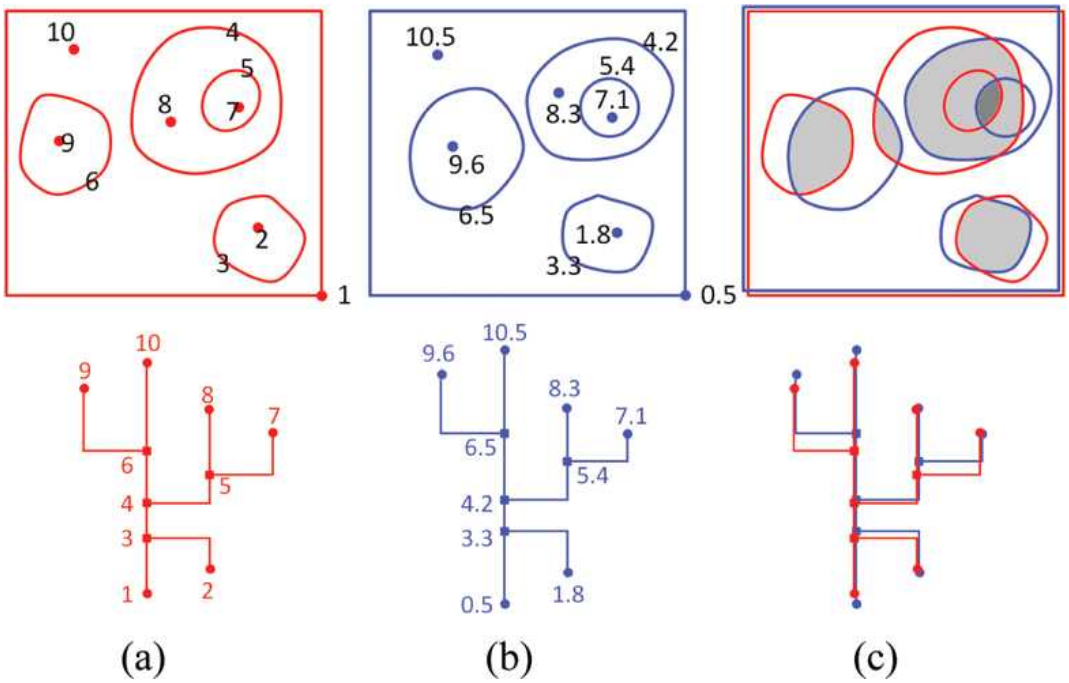

Figure 9: Branch correspondence indicated by contour region overlaps The nodes and contours are numbered by their iso-values. (a) and (b) show contour trees of two data fields and the corresponding contour regions of the tree branches (c) The matched branches are indicated by the shared x-axis locations

3.2.3.2 Topology-Level Uncertainty Visualization via Contour Tree Mapping

We map the contour tree of each ensemble member to the mean contour tree hierarchically. In the resulting visualization, the contour trees of different ensemble members are rendered beneath the mean contour tree in different colors. The *x*-axis locations of the matched branches are aligned. The number of unmatched branches for a branch in the mean contour tree is encoded as the width of the branch.

As shown in Figure 10, the mapped contour trees indicate how uncertain the iso-value ranges of individual branches are and how uncertain the number of branches is. The places where branches of the contour trees disagree with each other indicate the uncertain iso-value ranges of different contour regions segmented by branches. The overall blurring or clearness of the display indicates the overall high or low uncertainty. The thickness of a branch indicates the number of matched branches found in the ensemble members. For example, a thin branch (indicated by the green arrow) is found in the lower right corner of the tree. It represents an uncertain contour region (or minimum) which only appear in a few ensemble members. As shown in Figure 10, only two contours (in blue and green) from ensemble members 4 and 7 are shown after clicking on the branch. On the contrary, the two sets of contours selected from the middle locations of two thick branches (indicated by red and blue arrows) include corresponding contours from all the ensemble members. Accordingly, the topology-level uncertainty provides users a quick and high level overview of how much uncertainty there is in the topology of contours.

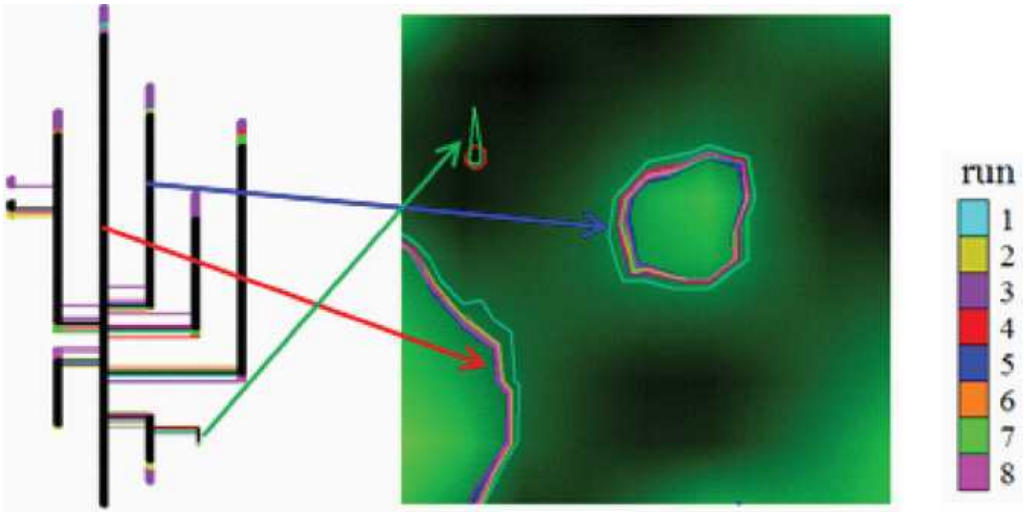

Figure 10: Topology-level uncertainty visualization Three sets of contours are shown after clicking on three locations (indicated by arrows) in the contour tree

3.3 User Interface Design

The contour tree provides an intuitive interface for exploring uncertainty. Figure 11 shows the interface designed to enable efficient browsing, manipulation, and quantitative analysis of uncertain scalar data fields. The top-left area shows a 2D or 3D visualization of a data set, including iso-lines and color-mapped image for 2D data, or iso-surfaces and volume rendering for 3D data. It provides interactions such as rotation and zooming. The top-right area shows a 2D display of the contour tree of the mean data field. It allows interactive contour tree simplification and contour selection. The three bottom areas show the data-level uncertainty, contour variability, and topology variability of the data based on contour trees. They allow similar interactions as the top-right contour tree display area. Once a data set is imported with the pre-computed uncertainty and variability information, the tool shows the color-mapped data (volume rendered for 3D data) in the data display area. The contours (iso-lines or iso-surfaces) are also rendered in the data display area and updated accordingly as a user clicks on one contour tree or changes the iso-value by sliding on the vertical bar in the top-right contour tree display area.

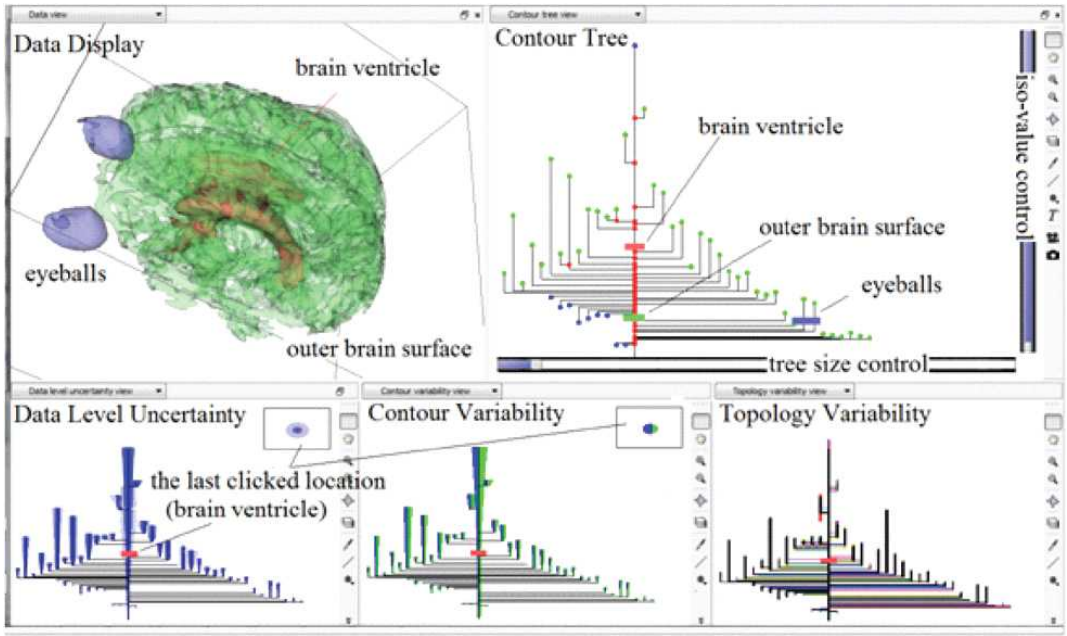

Figure 11: The user interface Top-left: data display area shows iso-surfaces of the mean field of five brains chosen from the simplified mean contour tree on the top right Top-right: contour tree display area shows a simplified contour tree Bottom: display areas for data-level uncertainty, contour-level uncertainty, and topology-level uncertainty shown on simplified contour trees

Users may simplify the contour tree in two ways. (1) Use a horizontal bar under the contour tree display to control the current number of branches in the contour tree. As a user drags the slider from right to left, branches are removed from the contour tree according to the order stored in the pre-computed simplification sequence. (2) Directly right-click on the inner nodes of the contour tree to prune or extend sub-trees.

Users may select contours to display in two ways. (1) Display multiple contours of the mean field with selected iso-values by clicking on the contour tree of the top-right area. A selected contour is shown in the same distinct color with a point placed at the clicked location. (2) Display a set of corresponding contours (spaghetti plots) with the same iso-value in different ensemble members in the data display area by double-

clicking on a location in one of the three contour trees in the bottom. Contours from different ensemble members are shown in different colors.

3.4 Application and Discussion

In this section, we apply the new uncertainty visualization to a simulated weather data set (135×135×30) and a medical volumetric data set (128×128×71).

The first experiment demonstrates an effective application of the new uncertainty visualization on the simulated data from Weather Research and Forecasting (WRF) model runs. The members of numerical weather prediction ensembles are individual simulations with either slightly perturbed initial conditions or different model parameterizations. Scientists use the average ensemble output as a forecast and utilize spaghetti plots to analyze the spread of the ensemble members. In our application, water-vapor mixing ratio data from eight simulation runs of the 1993 Super storm are used. Figure 12a shows the original contour tree (1734 critical points and 867 branches) and all the contours with the iso-value indicated by the red horizontal line connecting the slider of the vertical bar. Uncertainty information is shown in the simplified contour tree in Figure 12c. A few thin branches appear at its bottom region indicating the uncertain contour region (or minima) in the data. Figure 12b shows the volume rendering of the average data with circular uncertainty glyphs whose sizes vary according to the level of data-level uncertainty. Figure 12d shows a set of corresponding contours with high data-level uncertainty while Figure 12e shows a set of corresponding contours with high contour variability. Circular graduated glyphs indicating the magnitude of the data-level uncertainty and contour variability are shown at the bottom. The color bar on the right indicates the correspondence between the iso-surfaces and simulation runs.

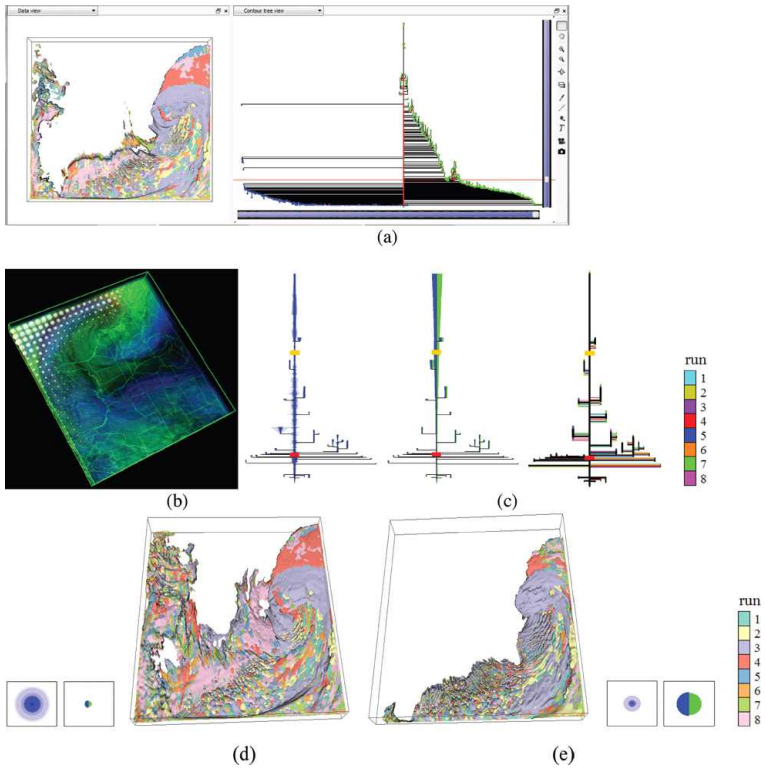

Figure 12: Exploration of weather ensemble simulation with contour tree based visualization (a) Original contour tree (b) Volume rendering with uncertainty glyphs (c) Left to right: data-level uncertainty, contour variability, and topology variability shown in the simplified contour tree (d) A set of contours (corresponding to the red points in the contour trees) with high data-level uncertainty indicated by the large graduated circle (e) A set of contours (corresponding to the yellow points in the contour trees) with high contour variability indicated by the large glyph for contour variability

The second application is to visualize non-weighted diffusion images. We apply our method to study the variation between the brain images from five subjects after affine registration to one particular data set chosen at random using FLIRT (Jenkinson & Smith, 2001). Contour trees had been introduced to explore brain data in previous literature (Hamish Carr & Snoeyink, 2003). The between-subject variation in brain anatomy is closely related to the uncertainty study of brain imaging (Eickhoff et al., 2009). The mean field of the brains contains 2143 critical points and 1071 branches in the original contour tree. Figure 13a shows the integrated visualization of both data and uncertainty glyphs. Variability information is shown in the simplified contour tree in Figure 13b. Brain outer surfaces and ventricle surfaces are selected from two points (indicated in red and yellow) in the contour trees. As shown in Figure 13a, the inner part of the brain area exhibit higher data value along with higher data-level uncertainty. This is reflected in the generally thicker graduated ribbons observed in the upper region of the left tree in Figure 13b. Meanwhile, contours corresponding to the upper region of the trees have higher contour variability and higher topology variability, indicating the impact of the data-level uncertainty on the contours and contour trees. The inner features such as brain ventricles exhibit higher variability than the brain outer surfaces based on the quantified uncertainty and variability information shown on the corresponding location in the contour trees. The interactive visualization supports the exploration of variability information that is hidden in the conventional view for selected anatomical structures.

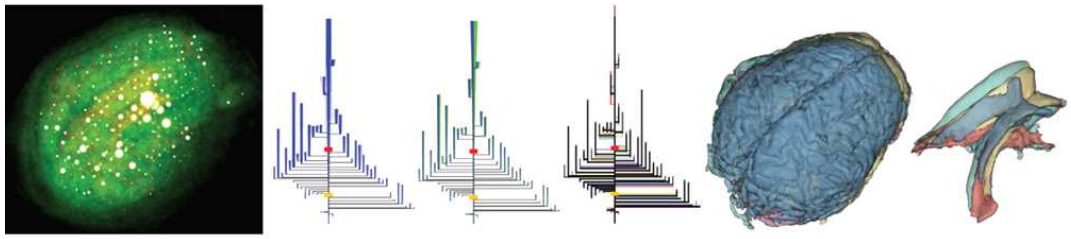

Figure 13: Brain data with uncertainty (a) Volume rendering with circular uncertainty glyphs (b) Left to right: data-level uncertainty, contour-level uncertainty, and topology-level uncertainty shown in a simplified contour tree (c) Corresponding contours of the same iso-value in different brain data. Five overlapped brain outer surfaces corresponding to the yellow points in the contour trees and five overlapped ventricle surfaces corresponding to the red points in the contour trees

As shown in the above examples, the new contour tree based visualization provides a combined exploration of both the data and the uncertainty. Users are allowed to look into three levels of uncertainty in the data. With the planar contour tree layout, users are allowed to explore data hierarchically and investigate the contours with certain uncertainty or variability more precisely and efficiently. Volume rendering with circular uncertainty glyphs (Figure 12d and Figure 13a) are provided for a comparison purpose. With 3D glyphs placed within the volumetric data, the details of the data are noticeably blocked by the glyphs while the glyphs are overlapped with each other or appear to be blurred or buried in a 3D scene. Accordingly, the new visualization method provides an alternative which shows both data and uncertainty clearly.

4 A FEATURE-BASED FRAMEWORK FOR VISUALIZING VECTOR FIELD UNCERTAINTY

In this section, we present a framework to visualize feature-level uncertainties in vector fields. In many cases, locations of features matter more than the data-level uncertainty. For instance, the locations of warm eddies are important in ocean fishery, and the locations of hurricane eyes or the peaks in pressure field are important in weather analysis. Two features are considered the same feature at in different data sets if they share the same tracking path (Tricoche, 2002) or the biggest similarity (Sohn & Chandrajit, 2006). We evaluate the uncertainty related to features by measuring the deviation of those feature pairs in different data sets. The features currently studied by us are vector field critical points. They are intuitive features closely related to physical features (Garth & Tricoche, 2005). Scalar fields can be analyzed through their gradient fields.

This section is organized as follows. Section 4.1 gives the method framework. Section 4.2, 4.3, and 4.4 discuss feature extraction, uncertainty measurement, and uncertainty glyph design respectively. Section 4.5 demonstrates and discusses application results.

4.1 Method Pipeline

The impact of uncertainty on the features is quantified as feature-level uncertainty which is measured by feature deviation Figure 14). The feature deviation is obtained through a three-step procedure—feature identification, feature mapping, and uncertainty representation. Given a set of data members (e.g. multiple simulation runs) this method first identifies the features within all the data members and the mean field given by averaging all the members. Second, feature tracking is implemented to map the features of each data member to that of the mean field. The mapped features are then assumed to have the same feature with slight position deviation in each of the individual data members. Finally, the feature-level uncertainties are expressed as the deviations of the features.

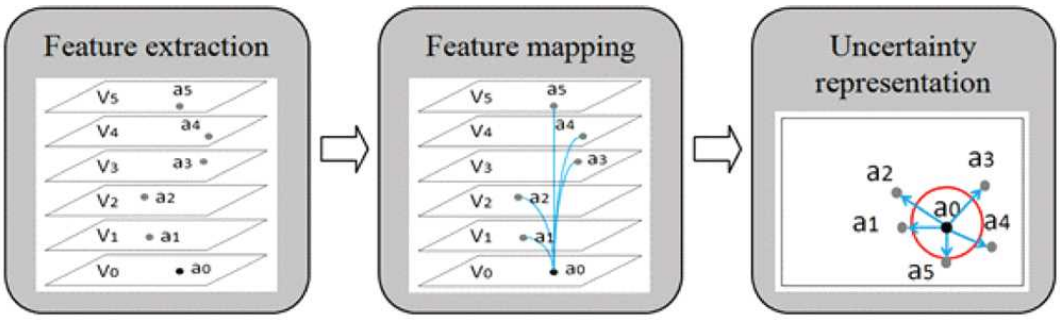

Figure 14: The pipeline for feature-level uncertainty visualization

4.2 Feature Extraction

Topology consists of critical points, periodic orbits, and separatrices. It characterizes a flow in that the relatively uniform flow behaviour in each topological region can be deduced from its boundary. The computation of critical points in a vector field can be found in (Helman & Hesselink, 1989). For a scalar field, its gradient field can be used to extract critical points so that the features of scalar fields and vector fields are analyzed in the same way. A vector field V can be constructed out of scalar field f using the gradient operator $\nabla : V = \nabla f = (\partial f / \partial x, \partial f / \partial y)$ The maxima of f appear as sinks and minima appear as sources in its gradient field V Figure 15 illustrates an example of the critical points extracted from the gradient field of a temperature field.

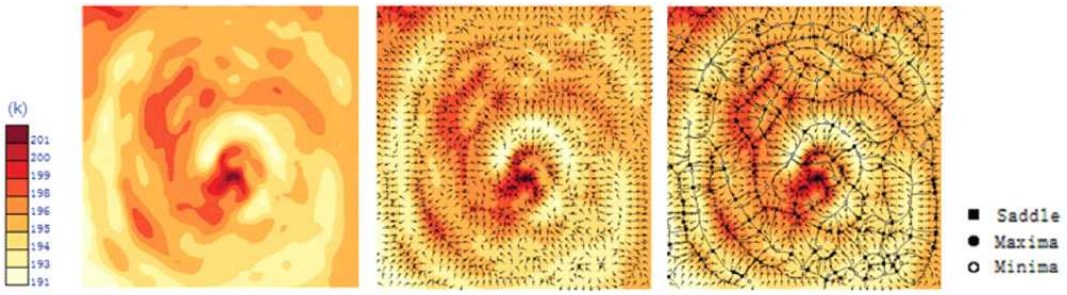

Figure 15: Topology extracted from the gradient field of a temperature field (a) Color-mapped temperature (b) Gradient field represented with arrows (c) Gradient field with extracted vector topology

4.3 Feature-Level Uncertainty Metric Based on Feature Mapping

Feature Flow Field (FFF) (Theisel & Seidel, 2003) is adopted to couple critical points of different fields by tracing streamlines within it. The concept of FFF has been successfully applied to tracking critical points (Theisel, Weinkauf, Hege, & Seidel, 2005), extracting Galilean invariant vortex core lines (Sahner, Weinkauf, & Hege, 2005), simplification (Theisel, Rössl, & Seidel, 2003a), and comparison (Theisel, Rössl, & Seidel, 2003b). In this chapter, it is used to identify the same feature that appears in different data members at different positions. The uncertainty of this feature is then expressed as the deviation between all of its counterparts in different data members.

Given *k* data members $V_i,(i=1,\ldots,k)$ a mean field V_0 is first computed as the average of them. V_0 is then paired with each data member V_i. Feature mapping is implemented for each data pair V_0 and V_i With the FFF method, features of different vector fields could be correlated.

Figure 16 demonstrates how to measure feature-level uncertainty related to a feature. For a data member V_1 and the mean field V_0 we trace a streamline from a critical point a_0 in V_0 until it reaches a critical point a_1 in V_1. After tracing critical points between all the pairs, the feature-level uncertainty is measured by the distances between $a_i(i=1,\ldots,k)$ and a_0 Figure 16a illustrates the feature mapping between a pair of data. Figure 16b shows a straightforward representation of the uncertainties related to individual features by arrows. Given a data member $V_n(1\le n\le k)$ it is possible that the streamline starting from a_0 reaches the boundary of the FFF or ends at V_0 instead of reaching a critical point in V_n In these cases, we assume that the mapped critical point a_n for a_0 in this data member exists outside the domain. Therefore, we set its distance from a_0 a large value — the maximum distance found between a_0 and all the mapped critical points a_i

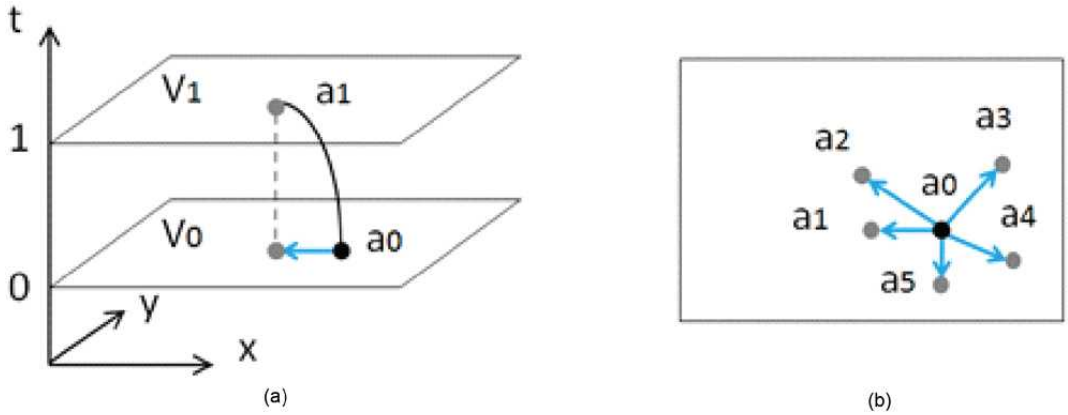

Figure 16: Feature-level uncertainty measurement (a) Feature deviations detected by tracing critical points within FFF between a data member V_1 and the mean field V_0 (b) Feature-level uncertainty measured by the deviations (indicated by arrows) of features (indicated by dots) in all the data members $V_n(n=1,\ldots,k$ from the features in the mean data field V_0

4.4 Uncertainty Representation

Glyph design addresses the central problem of how uncertainty information is processed into knowledge. For the detected deviations of a critical point, a quantitative glyph is designed to indicate uncertainty level related to the critical point. The new uncertainty glyph is inspired by both graduated circular glyph (Sanyal et al., 2010) and elliptical glyph (Walsum, Post, Silver, & Post, 1996).

4.4.1 Elliptical Glyph

A glyph that can be used at different levels is the elliptical glyph (Walsum et al., 1996). It depicts the covariance between multiple real-valued

random variables X_i. In probability theory and statistics, covariance measures how much two variables change together. The covariance matrix Σ generalizes the notion of variance to multiple dimensions. $\Sigma_{i,j} = Cov(X_i, X_j) = E[(X_i - E(X_i))(X_j - E(X_j))]$ where $E(X_i)$ is the expected value of X_i and the ellipse axis length is $\lambda = eig(\sum)$ Ellipse axis directions are given by $d = eigvec(\sum)$ An elliptical glyph can be applied to visualize tensors, but can also show a simplified representation of the spatial distribution of a set of 2D or 3D data (Sadarjoen & Post, 2000).

4.4.2 Graduated Elliptical Glyph

Before using the graduated ellipse, we considered using arrows to indicate the uncertainty by showing directions and distances of individual deviations. Nevertheless, it is found that arrows, though showing the deviations in an authentic way, could cause severe information overload when the number of ensemble runs increases. Contrarily, the graduated ellipse possesses the elliptical glyph's ability to depict the overall deviation of a feature and the graduated glyph's intuitive way to depict inner deviation of individual ensemble members. When placed across an image, the overall size and orientation of the glyphs indicate the variability of features while the very shape of and color distribution within an individual glyph give a quick statistical summary of uncertain deviations of a feature.

Let $a_0(x_0, y_0)$ be a critical point in the mean data field V_0 Its counter-parts in data members V_i are $a_i\ (x_i, y_i)(i = 1, \ldots, k)$ A graduated elliptical glyph consists of *k* nested ellipses and is placed at the location of a_0 The nested ellipses share the same orientation and axis ratio. The rendering of each nested ellipse is as follows:

First, assign an ellipse *E* with axes *A* and *B* computed according to the relative locations of a_i towards a_0, $(x_i - x_0, y_i - y_0)$ Let variable *X* be $x_i - x_0$ and *Y* be $y_i - y_0$ The lengths and directions of *A* and *B* are given by $eig(\sum(X, Y))$ and $eigvec(\sum(X, Y))$ respectively. $\sum(X, Y)$ is the Covariance Matrix of *X* and *Y*. Second, nested ellipses are produced by fitting them into ellipse *E*. Sort a_i according to its distance from a_0, $d_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}$ In descending order. Let the biggest distance among d_i be *D*. Each nested ellipse is given axis lengths $|A| \times d_i / D$ and $|B| \times d_i / D$ Finally. In a way similar to producing a graduated circular glyph (Sanyal, et al., 2010), assign saturation level s_i for the *i*th glyph as $s_i = (i - 1) / (k - 1)$ and overlay the ellipse so that the smaller one is over the larger one.

Figure 17 gives a comparison between using a simple ellipse, arrows, graduated circular glyph, and graduated elliptical glyph for feature-level uncertainty. The individual features $a_i, (i = 1, \ldots, k)$ (which are not displayed in a final visualization) are shown as well to better illustrate the design concept of each glyph. The simple ellipse (Figure 17a) only characterizes the overall deviation of a feature. Arrows (Figure 17b) indicate the exact locations of all the deviated features while inviting visual clutters when *k* increases. The graduated circular glyph (Figure 17c) shows the individual deviation of a feature but no direction information is revealed. However, the graduated elliptical glyph (Figure 17d) summarizes overall and individual distribution of a feature in a succinct way. Figure 18 shows a set of graduated ellipse with varying color distributions, sizes, axis length ratios, and orientations.

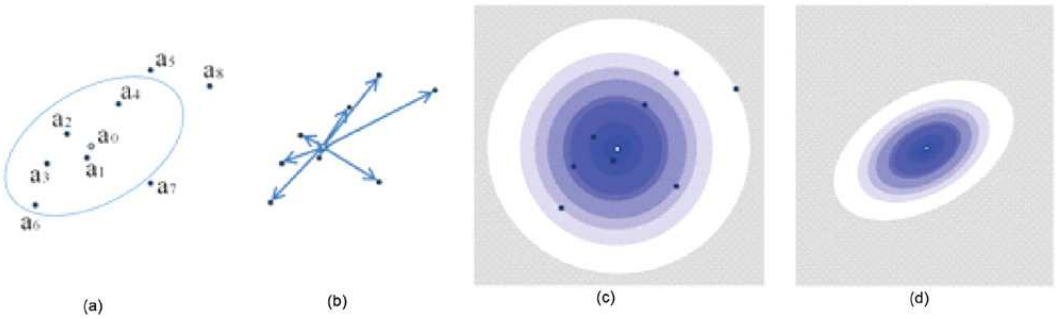

Figure 17: Feature-level uncertainty glyph design (a) Ellipse (b) Arrows (c) Graduated circular glyph (d) Graduated ellipse

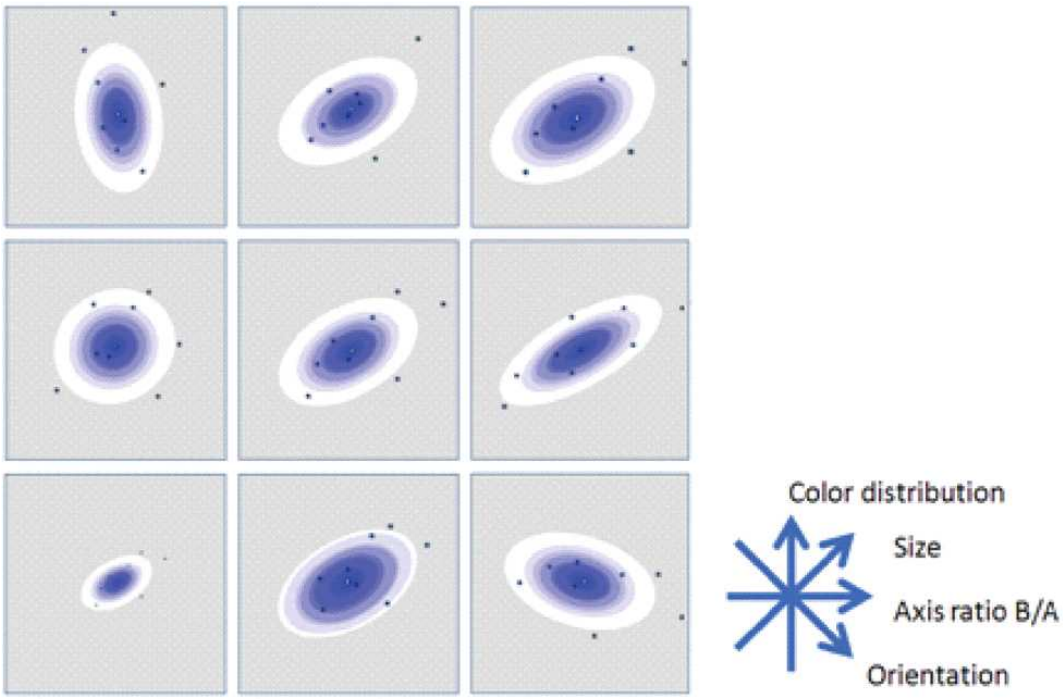

Figure 18: Graduated ellipses with varying orientation, saturation distribution, size, and axis length ratio. *B / A* 8 data members are used

4.5 Application and Discussion

The method is applied to a 2D vector dataset. The first dataset includes 5 simulated hurricane wind fields (Figure 19). The uncertainty glyphs are placed in critical point locations. The results demonstrate how much features are affected by the uncertainty within the data. Through mapping and comparing the critical points between different ensemble members, the shifts between critical points become perceivable. The graduated elliptical glyphs effectively indicate the magnitude and overall orientation of the uncertain deviations of vortices. The uncertain position of the hurricane eye reflects the impact of the uncertainty directly. A side-by-side display of different component data or the visualization of the data-level uncertainty may not give viewers such insight.

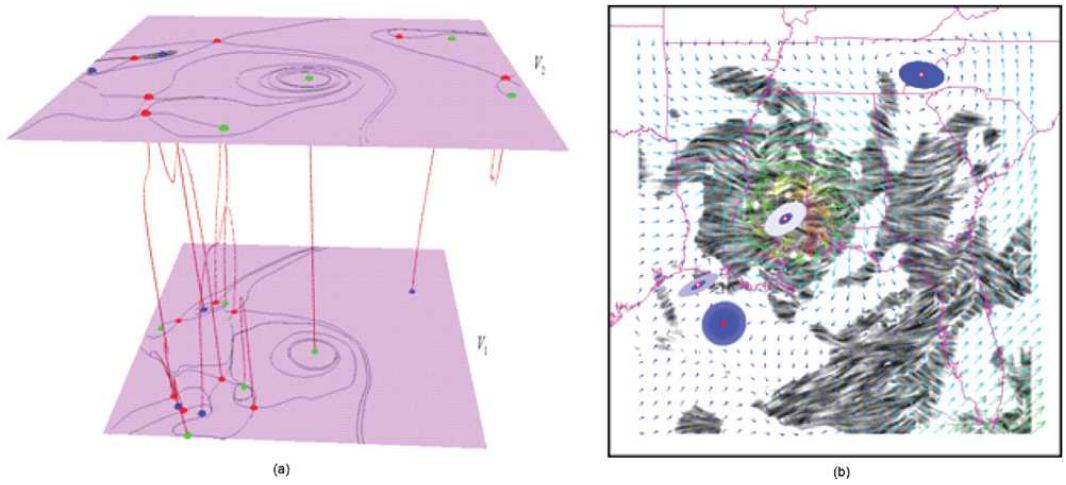

Figure 19: Feature-level uncertainty of a hurricane wind field (a) Feature tracking within FFF of two vector fields V_1 and V_2 (b) Uncertain location of hurricane eye and vortices in a hurricane wind field (5 ensemble members)

Although the result of this feature-level uncertainty visualization is positive, there are a few limitations and areas that need further study. Most notably, more features could be considered in the future. Second, other feature-mapping methods may be included depending on the feature type since the current feature-mapping method, FFF, mainly tracks topological features.

5 CONCLUSION

This chapter has conducted an in-depth investigation of feature-level uncertainties and suggested a promising direction of future uncertainty studies in exploiting topology tools. The presented feature-based techniques alleviate the inherent perception issues such as clutter and occlusion in uncertainty visualizations in 3D or large 2D scenes. The incorporation of the feature-level uncertainties into visualization provides

insights into the reliability of the extracted features which otherwise would remain unknown with the visualization of only data-level uncertainty. In addition, the novel use of contour trees provides an effective solution for interacting with 3D or large 2D data sets with uncertainty.

There are many possible directions in which one could extend this work. (1) Extend the developed feature-based uncertainty visualization framework to study uncertainty in various fields. (2) Improve the interactive uncertainty visualization based on user feedbacks. (3) Investigate more metrics to measure uncertainty or variability and apply our methods to address different types of uncertainty models. (4) The possibilities inherent in topology tools are not exhausted yet. For example, the possibility of using Morse-Smale complex (Smale, 1961) for uncertainty study has not been investigated. Therefore, we will continue the current work in utilizing topology tools to visualize uncertainty.

REFERENCES

Bajaj, C. L., Pascucci, V., & Schikore, D. R. (1997). The contour spectrum . In *Proceedings of Visualization '97*. New Brunswick, NJ: IEEE Press.

Brodlie, K., Osorio, R. A., & Lopes, A. (2012). A review of uncertainty in data visualization . InDill, J, Earnshaw, R, Kasik, D, Vince, J, & Wong, P. C. (Eds.), *Expanding the Frontiers of Visual Analytics and Visualization* (81-109). Berlin: Springer. doi:10.1007/978-1-4471-2804-5_6

Carr, H. (2004). *Topological Manipulation of Isosurfaces.* (PhD Thesis). Vancouver, BC, Canada, University of British Columbia.

Carr, H., & Snoeyink, J. (2003). *Path seeds and flexible isosurfaces using topology for exploratory visualization*. In *Proceedings of the Symposium on Data Visualisation.* New Brunswick, NJ: IEEE Press.

Carr, H., Snoeyink, J., & Axen, U. (2003). Computing contour trees in all dimensions. *Computational. Geometry*. *Theory & Applications*, *24*(2), 75–94.

Carr, H., Snoeyink, J., & Panne, M. V. D. (2004). Simplifying flexible isosurfaces using local geometric measures . In *Proceedings of Visualization '04*. New Brunswick, NJ: IEEE Press. doi:10.1109/VISUAL.2004.96

Cedilnik, A., & Rheingans, P. (2000). Procedural annotation of uncertain information . In *Proceedings of Visualization '04*. New Brunswick, NJ: IEEE.

Davis, T. J., & Keller, C. P. (1997). Modeling and visualizing multiple spatial uncertainties. *Computers & Geosciences*, *23*(4), 397–408. doi:10.1016/S0098-3004(97)00012-5

Diggle, P., Heagerty, P., Liang, K.-Y., & Zeger, S. (2002). *Analysis of longitudinal data*. New York: Oxford University Press.

Djurcilova, S., Kima, K., Lermusiauxb, P., & Pang, A. (2002). Visualizing scalar volumetric data with uncertainty. *Computers & Graphics*, *26*(2), 239–248. doi:10.1016/S0097-8493(02)00055-9

Eickhoff, S. B., Laird, A. R., Grefkes, C., Wang, L. E., Zilles, K., & Fox, P. T. (2009). Coordinate-based activation likelihood estimation meta-analysis of neuroimaging data: A random-effects approach based on empirical estimates of spatial uncertainty. *Human Brain Mapping*, *30*(9), 2907–2926. doi:10.1002/hbm.20718

Garth, C., & Tricoche, X. (2005). Topology-and feature-based flow visualization: Methods and applications. In *Proceedings of the SIAM Conference on Geometric Design and Computing.* New York: ACM Press.

Grigoryan, G., & Rheingans, P. (2004). Point-based probabilistic surfaces to show surface uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, *10*(5), 564–573. doi:10.1109/TVCG.2004.30

Heine, C., Schneider, D., Carr, H., & Scheuermann, G. (2011). Drawing Contour trees in the plane. *IEEE Transactions on Visualization and Computer Graphics*, *17*(11), 1599–1611. doi:10.1109/TVCG.2010.270

Helman, J., & Hesselink, L. (1989). Representation and display of vector field topology in fluid flow data sets. *IEEE Computer Graphics and Applications*, *22*(8), 27–36.

Hengl, T., & Toomanian, N. (2006). Maps are not what they seem: Representing uncertainty in soil-property maps. In *Proceedings of 7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences.* Edgbaston, UK: World Academic Press.

Jenkinson, M., & Smith, S. (2001). A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, *5* (2), 143–156. doi:10.1016/S1361-8415(01)00036-6

Keqin, W., Zhanping, L., Song, Z., & Moorhead, R. J. (2010). Topology-aware evenly spaced streamline placement. *IEEE Transactions on Visualization and Computer Graphics*, *16*(5), 791–801. doi:10.1109/TVCG.2009.206

Kosara, R., Miksch, S., Hauser, H., Schrammel, J., Giller, V., & Tscheligi, M. (2002). *Useful Properties of Semantic Depth of Field for Better F+C Visualization*. In *Proceedings of the Symposium on Data Visualisation.* New Brunswick, NJ: IEEE Press.

MacEachren, A. M. (1992). Visualizing uncertain information. *Cartographic Perspective*, *13*(3), 10–19.

Otto, M., Germer, T., Hege, H.-C., & Theisel, H. (2010). Uncertain 2D vector field topology. *Computer Graphics Forum*, *2*(29), 347–356.

doi:10.1111/j.1467-8659.2009.01604.x

Pang, A., Wittenbrink, C., & Lodha, S. (1997). Approaches to uncertainty visualization. *The Visual Computer*, *13*(8), 370–390. doi:10.1007/s003710050111

Pascucci, V., Cole-McLaughlin, K., & Scorzelli, G. (2004). Multi–resolution computation and presentation of contour tree. In *Proceedings of the IASTED Conference on Visualization, Imaging, and Image.* Calgary, AB: ACTA Press.

Pauly, M., Mitra, N. J., & Guibas, L. (2004). Uncertainty and variability in point cloud surface data. In *Proceedings of Eurographics Symposium on Point-Based Graphics.* New Brunswick, NJ: IEEE Press.

Pfaffelmoser, T., Reitinger, M., & Westermann, R. (2011). Visualizing the positional and geometrical variability of isosurfaces in uncertain scalar fields. In *Proceedings of Eurographics/IEEE Symposium on Visualization*. New Brunswick, NJ: IEEE Press.

Pöthkow, K., & Hege, H.-C. (2011). Positional uncertainty of isocontours: Condition analysis and probabilistic measures. *IEEE Transactions on Visualization and Computer Graphics*, *17*(10), 1393–1406. doi:10.1109/TVCG.2010.247

Rhodes, P. J., Laramee, R. S., Bergeron, R. D., & Sparr, T. M. (2003). Uncertainty visualization methods in isosurface rendering . In *Proceedings of Eurographics*. New Brunswick, NJ: IEEE Press.

Sadarjoen, I. A., & Post, F. H. (2000). Detection, quantification, and tracking of vortices using streamline geometry. *Computers & Graphics*, *24*(3), 333–341. doi:10.1016/S0097-8493(00)00029-7

Sahner, J., Weinkauf, T., & Hege, H.-C. (2005). Galilean invariant extraction and iconic representation of vortex core lines . In *Proceedings of EuroVis*. New Brunswick, NJ: IEEE Press.

Sanyal, J., Zhang, S., Bhattacharya, G., Amburn, P., & Moorhead, R. (2009). A user study to compare four uncertainty visualization methods for 1D and 2D datasets. *IEEE Transactions on Visualization and Computer Graphics*, *15*(6), 1209–1218. doi:10.1109/TVCG.2009.114

Sanyal, J., Zhang, S., Dyer, J., Mercer, A., Amburn, P., & Moorhead, R. J. (2010). Noodles: A tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, *16*(6), 1421–1430. doi:10.1109/TVCG.2010.181

Schmidt, G. S., Chen, S.-L., Bryden, A. N., Livingston, M. A., Osborn, B. R., & Rosenblum, L. J. (2004). Multidimensional visual representations for underwater environmental uncertainty. *IEEE Computer Graphics and Applications*, *24*(5), 56–65. doi:10.1109/MCG.2004.35

Schneider, D., Wiebel, A., Carr, H., Hlawitschka, M., & Scheuermann, G. (2008). 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. doi:10.1109/TVCG.2008.143

Smale, S. (1961). On gradient dynamical systems. *The Annals of Mathematics*, *71*(1), 199–206. doi:10.2307/1970311

Sohn, B. S., & Chandrajit, B. (2006). Time-varying contour topology. *IEEE Transactions on Visualization and Computer Graphics*, *12*(1), 14–25. doi:10.1109/TVCG.2006.16

Takahashi, S., Takeshima, Y., & Fujishiro, I. (2004). Topological volume skeletonization and its application to transfer function design. *Graphical Models*, *66*(1), 24–49. doi:10.1016/j.gmod.2003.08.002

Theisel, H., Rössl, C., & Seidel, H.-P. (2003a). Combining topological simplification and topology preserving compression for 2D vector fields . In *Proceedings of Pacific Graphics*. New Brunswick, NJ: IEEE Press. doi:10.1109/PCCGA.2003.1238287

Theisel, H., Rössl, C., & Seidel, H.-P. (2003b). Using feature flow fields for topological comparison of vector fields . In *Proceedings of Vision, Modeling, and Visualization*. New Brunswick, NJ: IEEE Press.

Theisel, H., & Seidel, H.-P. (2003). Feature flow fields . In *Proceedings of Data Visualization*. New Brunswick, NJ: IEEE Press.

Theisel, H., Weinkauf, T., Hege, H.-C., & Seidel, H.-P. (2005). Topological methods for 2D time-dependent vector fields based on stream lines and path lines. *IEEE Transactions on Visualization and Computer Graphics*, *11*(4), 383–394. doi:10.1109/TVCG.2005.68

Tierny, J., Gyulassy, A., Simon, E., & Pascucci, V. (2009). Loop surgery for volumetric meshes: Reeb graphs reduced to contour trees. *IEEE Transactions on Visualization and Computer Graphics*, *15*(6), 1177–1184. doi:10.1109/TVCG.2009.163

Tricoche, X. (2002). *Vector and Tensor Field Topology Simplification, Tracking and Visualization.* (Dissertation). Kaiserslautern, Germany, University of Kaiserslautern.

Walsum, T. V., Post, F. H., Silver, D., & Post, F. J. (1996). Feature extraction and iconic visualization. *IEEE Transactions on Visualization and Computer Graphics*, *2*(2), 111–119. doi:10.1109/2945.506223

Wittenbrink, C., Pang, A., & Lodha, S. (1996). Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, *2*(3), 266–279. doi:10.1109/2945.537309

Wu, K., & Zhang, S. (2013). A contour tree based visualization for exploring data with uncertainty. *International Journal for Uncertainty Quantification*, *3*(3), 203–223. doi:10.1615/Int.J.UncertaintyQuantification.2012003956

KEY TERMS AND DEFINITIONS

Ensembles: Multiple predictions from an ensemble of model runs with slightly different initial conditions and/or slightly different versions of models. Forecasters use ensembles to improve the accuracy of the forecast by averaging the various forecasts.

Feature: Phenomena, structures, or objects of interest in a dataset. The definitions of features depend on specific applications and users. The examples of topological features in a scalar field are Morse-Smale complex, Reeb-graph, and contour tree. Generally, vector field topology consists of critical points, periodic orbits, and separatrices.

Feature-based Visualization: A type of visualization which visualizes features extracted from the original and usually large data set.

Uncertainty: The opposite of certainty, having limited knowledge to exactly describe an existing state and the future outcome, etc. Uncertainty in scientific data can be broadly defined as statistical variations, spread, errors, differences, and minimum maximum range values, etc.