Visualization of Input Parameters for Stream and Pathline Seeding

Tony McLoughlin[1]
Matt Edmunds[1]
Chao Tong[1]
Robert S Laramee[1]
and Ian Masters[1]
Swansea University, Wales, UK.[1]

Guoning Chen[2]
Nelson Max[3]
University of Houston, Texas, US.[2]
University of California, Davis, US.[3]

Harry Yeh[4]
and Eugene Zhang[4]
Oregon State University, US.[4]

Abstract—**Uncertainty arises in all stages of the visualization pipeline. However, the majority of flow visualization applications convey no uncertainty information to the user. In tools where uncertainty is conveyed, the focus is generally on data, such as error that stems from numerical methods used to generate a simulation or on uncertainty associated with mapping visualization primitives to data. Our work is aimed at another source of uncertainty - that associated with user-controlled input parameters. The navigation and stability analysis of user-parameters has received increasing attention recently. This work presents an investigation of this topic for flow visualization, specifically for three-dimensional streamline and pathline seeding. From a dynamical systems point of view, seeding can be formulated as a predictability problem based on an initial condition. Small perturbations in the initial value may result in large changes in the streamline in regions of high unpredictability. Analyzing this predictability quantifies the perturbation a trajectory is subjugated to by the flow. In other words, some predictions are less certain than others as a function of initial conditions. We introduce novel techniques to visualize important user input parameters such as streamline and pathline seeding position in both space and time, seeding rake position and orientation, and inter-seed spacing. The implementation is based on a metric which quantifies similarity between stream and pathlines. This is important for Computational Fluid Dynamics (CFD) engineers as, even with the variety of seeding strategies available, manual seeding using a rake is ubiquitous. We present methods to quantify and visualize the effects that changes in user-controlled input parameters have on the resulting stream and pathlines. We also present various visualizations to help CFD scientists to intuitively and effectively navigate this parameter space. The reaction from a domain expert in fluid dynamics is also reported.**

Keywords—*Seeding Parameter Sensitivity, Uncertainty, Exploratorive Visualization.*

I. Introduction

Visualization of uncertainty is identified as one of the top future visualization problems [1], [2]. Uncertainty may be introduced in various stages of the visualization pipeline (Figure 1). Uncertainty may arise from numerical error and visualization parameter instability. For example, in the data acquisition phase, measurement instruments are associated with an inherent amount of error. Uncertainty is also introduced in simulations by both the discrete PDEs (partial differential

equations) used in the computation and from the discrete grid-based mesh used to save the result. The mapping of data to visualization primitives may also introduce uncertainty. For example, the use of piecewise linear interpolation to reconstruct data introduces error. Streamline integrators accumulate error along the length of their curves. However, discrete primitives such as streamlines and iso-surfaces imply certainty.
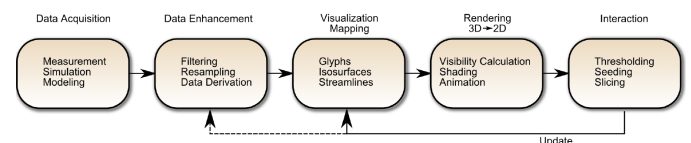

Fig. 1: The visualization pipeline. Each stage in the pipeline contains uncertainty. Our goal is to identify, quantify and visualize the uncertainty associated with user-input (right) typical for flow visualization and to present it to the user in a helpful and meaningful way.

Another source of uncertainty stems from user-input parameters. Small changes to input parameters can introduce large changes to the resulting visualization. In the context of dynamical systems, a seed position represents an initial condition and the trajectory that a stream or pathline takes can be predicted [3]. However, the reliability or certainty of the prediction fluctuates. Small sub-pixel perturbations in seeding positions can sometimes result in unpredictable or uncertain behavior in the form of very different integral paths. This is clearly demonstrated by Chen et al. [4] [5]. They identify areas in the seeding domain according to high uncertainty associated with flow trajectories and take this error into account in their visualizations based on Morse Decomposition. In this paper, we visualize the predictability uncertainty associated with manual seeding directly in the space-time domain such that there is a direct coupling of the user-input parameter space and the sensitivity (or reliability) of predicting a stream or pathline's trajectory. We use the term *sensitivity* to describe the notion that a small change in user input value can cause a large change to visualization output. In other words there are certain regions of the domain where the user is less certain what the effects of varying seeding position are.

It is not clear to what extent the output of a flow visu-

alization technique is affected by the combined effect of the uncertainties introduced in the various stages of the visualization pipeline. Without conveying these uncertainties, any flow visualization may be misleading. Interestingly, a lot of focus is placed on interaction in the visualization literature. However very little attention is paid to the uncertainty introduced by user-input parameters and their corresponding threshold values - the last stage of the visualization pipeline. Furthermore, interaction may be iterative, e.g., visualization results are repeatedly refined. Conceptually, this paper advocates a visual map of user-input sensitivity.

We focus on uncertainty introduced and associated with user-controlled input parameters for flow visualization. While there have been a number of papers examining uncertainty in flow visualization due to the data acquisition and visualization phases of the pipeline, relatively little research has been invested into visualizing the effects and uncertainty due to interaction or user-input to flow visualization. In many cases the user may be unaware of effects due to input parameter stability. Specifically, we analyze the effect that stream and pathline seed position and seeding rake position, in both space, time and in orientation, have upon the resulting set of integral lines. Inter-seed spacing along a seeding rake is also investigated. Our investigation includes unsteady flow, highlighting regions within the entire spatio-temporal domain that are sensitive to stream and pathline seeding position and rake orientation. The goals of our investigation are to: 1) Investigate the sensitivity of flow visualization with respect to user-input parameters used for stream and pathline seeding position in both space and time. 2) Evaluate the stability of user-specified values for rake positioning and orientation. 3) Provide visualization tools to explore and investigate the behavior and sensitivity of stream and pathline seeding for different user-controlled input parameters. As part of our contributions we study and visualize the following:

- The sensitivity of stream and pathlines to seeding position in space and time and seeding rake position and orientation.

- A streamline similarity metric, including a comparison of various metrics, that can be used to quantify flow sensitivity, i.e., reflect areas where a small change in position results in large changes to streamline geometry.

- The sensitivity of inter-seed distance between adjacent streamline seeds, resulting in a novel method for distributing stream and pathline seeds along a rake.

- Novel techniques to visualize these user-input parameters and their associated sensitivity.

We also demonstrate our visualizations to a domain expert from fluid mechanics and report his feedback. With existing flow visualization tools, discovering how user-input parameters and threshold values affect the visualization results require what essentially amounts to a manual search through parameter space. The space may grow exponentially with each new user input option. A manual process is very time consuming and in some cases may even be impossible. Even if parameters can be modified interactively, users may get lost in their search through parameter space and the space-time domain.

To provide insight into a system's behavior requires specific visualization techniques for a range of user input.

The rest of the paper is organized as follows: Section II describes previous work. Section III details and evaluates metrics for computing stream and pathline seeding sensitivity. Section IV demonstrates our method applied to a range of user-input parameters including seeding position, rake position and rake orientation. Section V discusses the application of our method to adaptively spacing seeds along a seeding rake. Section VI shows how seeding position can be visualized in unsteady flow, highlighting interesting regions in the spatio-temporal domain. An evaluation of performance is conducted in Section VII. Finally, the paper is concluded in Section VIII.

II.   RELATED WORK

Figure 1 illustrates the visualization pipeline. Uncertainty is accumulated at each stage. The goal of this work is to identify uncertainty associated with the parameters controlled by the user in the interaction stage (right).

A.   Data Uncertainty and Visualization Mapping Uncertainty

Wittenbrink et al. [6] investigate the use of glyphs to visualize uncertainty. A variety of glyphs are presented and evaluated, typically mapping uncertainty to glyph attributes such as length, area, color and/or angle.

Lodha et al. [7] present a system for visualizing uncertainty called UFLOW. UFLOW is used to analyze the changes resulting from different integrators and step-sizes used for computing streamlines. Visualization of the differences between the streamlines is achieved using several methods such as glyphs that encode uncertainty through their shape, size and color. A novel application of mapping sound to uncertainty is presented by Lodha et al. [8].

Cedilnik and Rheingans [9] simultaneously visualize the data and its uncertainty while minimizing the distraction that can occur when both are visualized together. A procedural texture incorporates a grid-like structure which is blended with the image. To represent the uncertainty the grid is deformed with the amount of deformation mapped to the uncertainty within a given region.

Rhodes et al. [10] visualize uncertainty in volume visualization. They extend the Marching Cubes algorithm so that an uncertainty value is associated with each sample. Uncertainty is mapped either to hue or to the opacity value of a stipple texture pattern on a secondary surface that envelops the iso-surface.

Pang et al. [11] and Verma and Pang [12], present comparative visualization tools to analyze differences between two datasets. Streamlines and streamribbons are generated on two datasets, one being a sub-sampled version of the other. To compare streamlines, Euclidean distance between them is used. Visualization primitives can then be added to aid the user in seeing how a pair of streamlines differ.

Brown [13] demonstrates the use of vibrations to visualize data uncertainty. Experiments using oscillations in vertex displacement, and changes in luminance and hue are investigated.

Botchen et al. [14] present a method of visualizing uncertainty in flow fields. Texture advection is used to provide a dense visualization of the underlying flow field. Error in texture-based representations of flow is handled by using a convolution filter to smear out particle traces, modifying the spatial frequencies perpendicular to the particle traces.

Sanyal et al. [15] depict uncertainty in numerical weather models using glyphs, ribbons and spaghetti plots. Praßni et al. [16] highlight areas of ambiguity and allow the user correct potential misclassification of volume segmentation.

Potter et al. [17] present an extension to the traditional box plot in order to portray uncertainty information.

B. User-input Uncertainty

Brecheisen et al. [18] analyze the stability of user-input parameters used for Diffusion Tensor Imaging fiber tracking, specifically the terminating criteria for the fiber tracking. Often, the same set of fiber tracking parameters is applied to several different data sets. Thus, the user may be unaware of what effect the current parameter configuration has on the current data. They visualize the effects of two common termination criteria for fiber tracking, namely, the anisotropy threshold and the curvature threshold. This is the inspiration behind the current work. Berger et al. [19] introduce an interactive approach to continuous analysis of a sampled parameter space. Using this approach users are guided to potentially interesting parameter regions. Uncertainty in the prediction is visualized using 2D scatterplots and parallel coordinates.

Hlawatsch et al. [20] augment mouse pointer seed and direction based on the largest valued regions of a Finite-Time Lypanov Exponent Field. We note that their study is limited to 2D fields only and does not focus on visualization of uncertainty itself, but rather controlling mouse interaction.

To the authors knowledge the present work is the only work on the visualization of user-input parameter sensitivity specifically for stream and pathline seeding and rake positioning. In fact Brecheisen et al. highlight seeding as a direction for future work. A recent survey covers the topic of streamline seeding and also uncertainty flow visualization [21]. The difference between our work and the above is that we are the first to focus on the actual seed positioning and rake input parameters, whereas Brecheisen et al. [18] focus on terminating criteria.

C. Manual Versus Automatic Streamline Seeding

There have been a number of important papers on automatic streamline seeding for 3D data. See for example Spencer et al. [22] and Li et al. [23]. See McLoughlin et al. [21] for a complete overview. However the focus of the work we present does not fall into this category. We classify these into the visualization mapping stage of the pipeline from Figure 1. In contrast, our work focuses on the last stage - interaction.

Another aspect we emphasize is the importance of manual seeding. CFD engineers require manual seeding. Interpreting the results of fully-automatic streamline seeding strategies without a complete understanding of the underlying algorithm is very difficult. In general, automatic seeding strategies do not take into account the full range of CFD attributes. Not all features of interest are known a priori. Thus feature-based streamline seeding has limitations.

D. Finite-Time Lyapunov Exponent

The Finite-Time Lyapunov exponent (FTLE) [24] has gained increasing attention of late [25]. It quantifies the local separation behavior of the flow. It is used to measure the rate of separation of infinitesimally close flow trajectories. It is computed from the set of all trajectories to produce a *flow map*, ϕ. Once the flow map has been computed, $\phi(\mathbf{x}_0, t_0, t)$ maps the position of the trajectory starting at time t_0 from position $\mathbf{x}_0$ for a finite-time, t. Using the flow map, the *Cauchy-Green deformation tensor field*, $\mathbf{C}_{t_0}^t$, t is obtained by left-multiplying the flow map with its transpose [26]:

$$\mathbf{C}_{t_0}^t(\mathbf{x}) = \left[\frac{\partial(\mathbf{x}_0, t_0, t)}{\partial(\mathbf{x}_0)}\right]^T \left[\frac{\partial(\mathbf{x}_0, t_0, t)}{\partial(\mathbf{x}_0)}\right] \qquad (1)$$

From this, the FTLE is computed by:

$$\mathbf{FTLE}_{t_0}^t(\mathbf{x}) = \frac{1}{t - t_0} \ln \sqrt{\lambda_{max}(\mathbf{C}_{t_0}^t(\mathbf{x}))} \qquad (2)$$

where $\lambda_{max}(\mathbf{M})$ is the maximum eigenvalue of matrix $\mathbf{M}$ [24].

FTLE requires the choice of a temporal window. The effect of a change in the time-window length has not been studied sufficiently [25]. A common use of FTLE is to extract *Lagrangian Coherent Structures* (LCS). LCS are extracted from an FTLE field by ridge extraction. This is dependent upon the definition of ridges themselves as well as upon the quality of the ridge extraction technique employed.

From a visualization point of view, FTLE is used to extract Lagrangian Coherent Structures and belongs to the visualization mapping stage of the pipeline (Figure 1). Standard feature-extraction algorithms involve four important stages: selection, clustering, attribute calculation, and iconic mapping [27]. Selection generally involves a search process through the data space, e.g., searching for minima or maxima. Then, filtered data is aggregated or clustered together, e.g., ridges or valleys are identified. Finally, icons are mapped to features, e.g., ridge or valley lines.

The work we present does not involve any explicit searching, or clustering and thus does not belong to the class of traditional topology or feature extraction algorithms. Our work focuses on the interaction stage and is not concerned with explicit extraction of flow features. The work presented here develops a visual map of streamline and pathline seeding parameter space.

III. A Method for Defining Streamline Seeding Sensitivity

We examine the effect of seeding position has on a streamline by analyzing and quantifying the change in curve geometry and direction when compared to another streamline seeded in close proximity. We describe spatial and temporal metrics for comparing the similarity of a pair of streamlines. Previous work on defining a similarity metric is introduced. This is followed by a description of the modifications we apply to the metric in order to incorporate an entire field of streamlines or pathlines. We discuss how the metric is used to create both spatial and temporal sensitivity field the key behind real-time navigation, exploration, and visualization of input parameter space. Figure 2 shows an overview of the workflow.
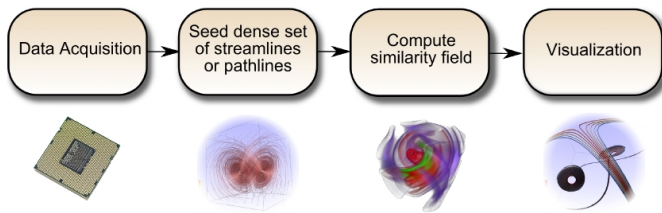
Fig. 2: An overview of our workflow used to visualize stream and pathline seeding and rake parameter sensitivity. A dense set of streamlines or pathlines are computed on the simulation data. This set is utilized to construct the sensitivity field in both space and time. Various visualization tools are employed to render the parameter-space based on the sensitivity field.
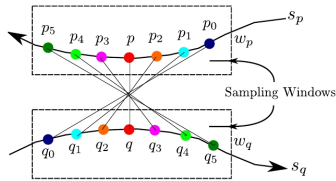

Fig. 3: Two sampling windows, w_p and w_q, placed on two streamlines, s_p and s_q. The sampling window incorporates the downstream direction and curvature in the computation of the similarity metric. The similarity between corresponding points along sub-lengths s_p and s_q is computed using Equation 3.

A. A Previous Streamline Similarity Metric

A streamline similarity metric is introduced by Chen et al. [28]. This approach (Figure 3) measures the similarity between local regions of streamlines in order to facilitate a streamline seeding strategy. For a given point, p, $p \in s_p$, the closest point, q, $q \in s_q$ is identified. Sampling windows are then placed over s_p and s_q. The sampling windows, w_p and w_q, sample n points on either side of p and q, about which they are centered (n is user defined). This yields two sets of evenly-spaced sample points, for s_p and s_q, $[p_0,...,p_{n-1}]$ and $[q_0,...,q_{n-1}]$. Note that these points retain the directional information of s_p and s_q. As we traverse from the 0th point to the $(n-1)^{th}$ point we proceed downstream. See Figure 3. These two sets of points are then used to compute the local similarity between s_p and s_q. The similarity, d is defined by:

$$d(s_p,s_q) = ||\overrightarrow{p-q}|| + \alpha \frac{\sum_{k=0}^{n-1} \left| ||\overrightarrow{p_k - q_k}|| - ||\overrightarrow{p-q}|| \right|}{n} \quad (3)$$

Equation 1 measures two key attributes. The first term measures the translational distance between s_p and s_q, while the second term corresponds to the shape and orientation difference. A weighting coefficient α is placed on the second term to allow the user to place more emphasis on the difference in shape. Values between 1 and 3 for α are found to be the most useful [28]. There are other possible similarity metrics, e.g., Hausdorff, for measuring curve similarity which we discuss in Section III-D.

B. An Adapted Streamline Similarity Metric

For computing streamline similarity, we adopt the above metric. However, there are some key modifications in order to
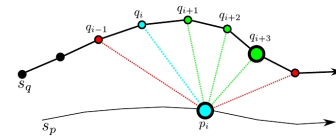

Fig. 4: Finding the associated point, q_n, on a neighboring streamline, s_q. The neighboring point is tested by computing the distance $d_i = ||p_i - q_i||$, marked in blue on the streamlines. We traverse in the downstream direction if d_i decreases. This is highlighted by the green points. This is repeated until d_i increases (the red points) at which point we store the current point, q_{i+3}. The same process is performed upstream. These two points (q_i and q_{i+3} in the example) are then compared and the pair with the shortest distance (the large green point in the figure) is stored.

compare entire streamlines to each other, not just subsections. Firstly, w_p and w_q, include the entire length of s_p and s_q. Every $p \in s_p$ is associated with a point along s_q. The algorithm for locating the associated point is presented below. Secondly, p_0 and q_0 are always seeding points. When both lists are complete the similarity distance, d, between s_p and s_q is computed. Note that this creates an asymmetry between the distance values. In general: $d(s_p,s_q) \neq d(s_q,s_p)$.

a) Associating Points between Streamlines and Shear:: Given a point p_i, $p_i \in s_p$, finding the associated point, q_j, $q_j \in s_q$, involves a search. Rather than a brute-force search, we start out by exploiting the parameterization along s_p and s_q. For the ith point along s_p, p_i, we start with the ith point, q_i, and compute the Euclidean distance $d_i = ||\overrightarrow{q_i - p_i}||$. If s_q contains fewer than i points, we simply compare the final point, i.e., q_{n-1} if s_q has n points. We test points in the downstream direction of s_q (i.e., q_{i+1}). If $d_{i+1} < d_i$, i.e., if $||p_i - q_{i+1}|| < d_i$, we then proceed to test q_{i+2}. This test is repeated until an increase in d is detected. Finally, we store the current point q_{min}. This is repeated in the upstream direction (starting from q_i). This results in two points. These may be the same point in the case that q_i is closer to p_i than q_{i+1} and q_{i-1}. We then determine which is closest to p_i. These points are then used in the similarity distance computation. This process is illustrated in Figure 4. The associated point is computed this way so that we compare corresponding local regions between s_p and s_q. This enables (1) the curvature, (2) translational distance, and (3) downstream direction to define the similarity between s_p and s_q.

Shear flow is another important property. This is highlighted in Figure 5. The user may be interested in factoring shear into the similarity metric. Thus we provide the user option to incorporate shear by using the corresponding points between s_p and s_q as resulting from their parameterization (Figure 5 right). If one streamline is longer than the other, the last point is used on the shorter streamline when its length has been traversed.

Incorporating shear into the similarity metric generally provides better results in our experiments. It also reduces computation time. Figure 5 compares two volume renderings of the sensitivity field without and with shear enabled. Enabling shear produces more well defined characteristics. Therefore,

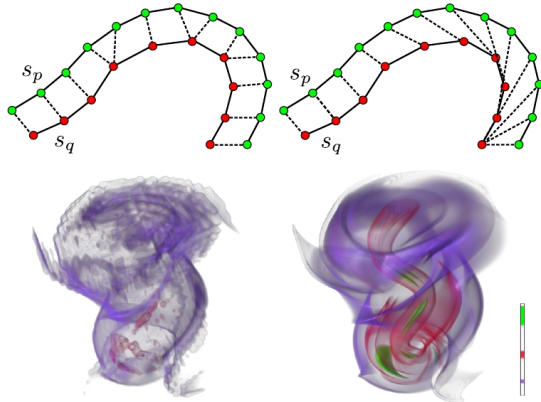we enable this option by default. All visualizations use this unless noted otherwise.

Fig. 5: (Top-Left) Connecting the associated points using our algorithm (Section III-B). This produces a method of comparison of s_p and s_q based on their geometry. The mean of the length of the connecting line segments is related to the similarity distance. (Top-Right) Connecting the points along s_p and s_q based on the parameterization of the streamlines. These connections form an alternative comparison of s_p and s_q. As s_p and s_q approach the curve the connecting lines become sheared and don't link the closest points between s_p and s_q measured by euclidean distance. If s_q has fewer points we use the last point. (Bottom) The spatial sensitivity fields created from a tornado simulation. The two images were generated without and with the shear attribute respectively.
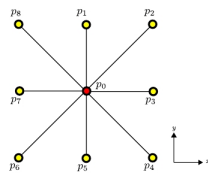

Fig. 6: A streamline, s, is seeded at every sample point, p_i, in the domain. In order to compute the sensitivity field for a given p_i, the similarity between $s(p_i)$ and its neighbors is computed. The mean of the similarity computations quantifies p_i (Eq. 4). This example is presented for a 2D field for simplicity.

C. Computing a Spatial Sensitivity Field

To facilitate interactive analysis, i.e., reporting parameter stability for a given set of seeding points we provide this information in real time to the user. We pre-compute a spatial sensitivity field that utilizes the above similarity metric for a dense set of streamlines. We formulate streamline seeding sensitivity in terms of the derived sensitivity field defined over the entire spatio-temporal domain.

For each sample position, p_i, within the domain, $\mathbb{R}^3$, a streamline, s_p, is computed. For every s_p within $\mathbb{R}^3$, the similarity is computed by incorporating each of its neighbors ($n = 8$ in 2D, $n = 26$ in 3D) (Figure 6). The total spatial

similarity value is computed using Eq. 4.

$$D(p_0) = \frac{\sum\limits_{i=0}^{n} d(s(p_0), s(p_i))}{n} \tag{4}$$

where, n is the number of neighbors and $d(s(p_0), s(p_i))$ is the similarity distance between a streamline seeded at p_0 and another seeded at p_i as in Eq. 2. Figure 6 illustrates this for a 2D field. See Figure 9 for a sample visualization of a sensitivity field.
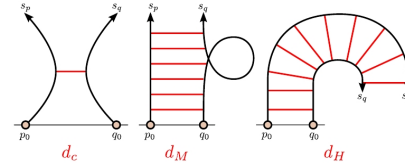
D. Candidate Metrics

Fig. 7: This figure shows some of the cases that inspire our choice of distance metric: d_c does not quantify curvature, d_M may not quantify curvature or shear, d_H does not quantify shear. Our metric measures (1) translational distance, (2) curvature, (3) downstream direction, and (4) shear.

In addition to the metric in Section III-B, we have experimented with various metrics for quantifying integral curve similarity. For example, the closest point distance, d_c, could be used:

$$d_c(s_p, s_q) = \min_{p_k \in s_p, q_k \in s_q} ||p_k - q_k|| \tag{5}$$

However d_c ignores curvature and shear. The mean distance $d_m(s_p, s_q)$ of closest distances is another candidate:

$$d_M(s_p, s_q) = \text{mean}(d_m(s_p, s_q), d_m(s_q, s_p)) \tag{6}$$
$$\text{where} \quad d_M(s_p, s_q) = \text{mean}_{p_k \in s_p} \min_{q_k \in s_q} ||p_k - q_k||$$

This does not quantify downstream direction or shear. The Hausdorff distance, d_H could be used:

$$d_H = \max(d_h(s_p, s_q), d_h(s_q, s_p)) \tag{7}$$
$$\text{where} \quad d_H(s_p, s_q) = \max_{p_k \in s_p} \min_{q_k \in s_q} ||p_k - q_k||$$

However, this does not intuitively quantify downstream direction and is computationally expensive. Another candidate is the Fréchet distance [29]:

$$d_f(s_p, s_q) = \min\left\{||L|| \;\; |L \text{ is an unordered coupling between } s_p \text{ and } s_q\right\} \tag{8}$$

However, this metric is computationally expensive and does not take shear into account. Interpretation by domain experts is also non-intuitive. Li et al. [30] also describe a similarity metric:

$$d_l = 1 - \frac{1}{2}\left(\frac{v'(p) \cdot v(p)}{|v'(p)| \; |v(p)| + 1}\right) \tag{9}$$

where v' is an approximate vector at p and $v(p)$ is the original sample vector. This metric compares vectors, not streamlines.

The Finite Time Lyapunov Exponent (FTLE) [24] is another candidate metric. FTLE is primarily a candidate for quantifying divergence (see Figure 8 and 9). We have done
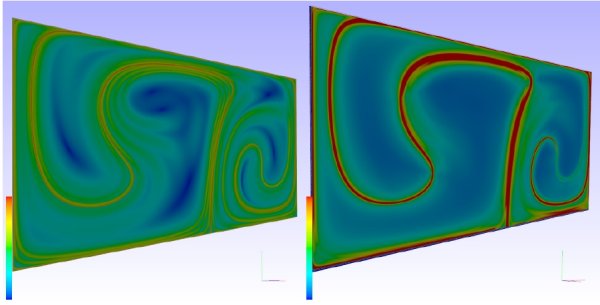
Fig. 8: (Left) The FTLE computed on a time-dependent double gyre simulation. (Right) Our sensitivity field derived on the same simulation.

some experiments to compare our metric with FTLE as shown in Figures 8 and 9. In general, they provide very similar results. Figure 9 better highlights the subtle differences between FTLE and the adapted metric from Chen et al. [28].

FTLE is suitable for the extraction of Lagrangian Coherent Structures (LCS) as stretching of the flow (divergence) often grows exponentially in time near LCS's. The LCS structures are derived by applying topological techniques to the FTLE field e.g., Ridge extraction [25].

The sensitivity field however is not only sensitive to divergence. Shear, and differences in curvature of neighboring streamlines are other characteristics which are inherently captured by this technique. The result of this approach can be seen in the differences between the top and bottom images of Figure 9. In the bottom image the effects of shear close to the centers of the vortex structures are highlighted, whereas the FTLE (top image) does not highlight these regions.

An observation is the apparent scaling of the data within the sensitivity field as compared to the FTLE field. For example, if the sensitivity data is modified similar to the FTLE computation e.g.,

$$\frac{1}{(t - t_0)} \ln \sqrt{\sum_{i=0}^{n} d(s(p_0), s(p_i))} \qquad (10)$$

then we see results very close to the FTLE visualization, while still highlighting the additional sensitivity information.

Another observation is that FTLE only considers the end points of the integral lines. The sensitivity field is computed as a sum of differences along the length of the integral lines, therefore it is possible to highlight regions with our technique where the flow separates and rejoins around an object in the path of the flow.

The choice of metric depends on the interests of the user. Figure 7 compares some metrics based on two streamlines, s_p and s_q seeded from a rake. It highlights special cases where flow properties are not quantified. We chose the metric described in Section III-B by default because it includes (1) downstream direction, (2) streamline curvature (not only distance), (3) divergence, and (4) shear. However, any streamline or pathline similarity metric can be used in our framework including FTLE. It is left as a user option. A full comparison

of different metrics is beyond the scope of this paper. We refer the reader to Moberts et al. [31] for a full evaluation.
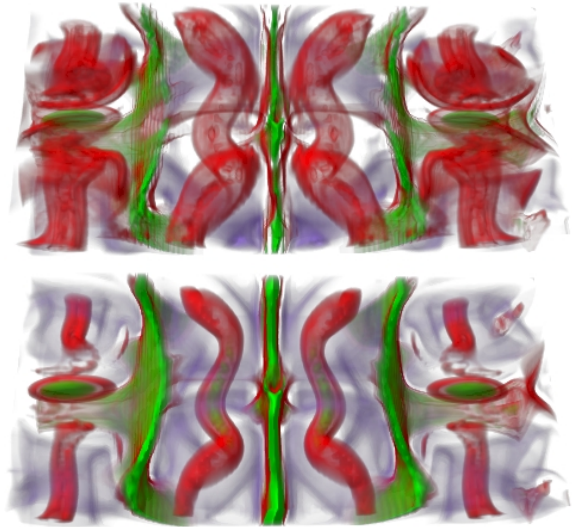

Fig. 9: (Top) FTLE based sensitivity field computed from the Bernard flow data with a streamline length of $t = 100$. (Bottom) Sensitivity field computed with a streamline length of $t = 100$. Both images are rendered using DVR with identical transfer functions.

IV. Visualization of Seeding Position and Rake Parameter Sensitivity

Our first approach to visualizing the sensitivity field directly is to derive a scalar field D (Eq. 4). This enables fast, intuitive exploration of data and provides the user a quick overview of regions with high rates of streamline-based change. With the spatial sensitivity field stored as a D, it is possible to visualize the sensitivity of the seed positions using any volume visualization technique.

Care must be taken by the user when interpreting the results of the sensitivity field. The initial impression is that the sensitivity field directly maps to features within the flow field. This is not necessarily the case. These regions may be displaced far away from any actual flow features. The sensitivity field is not a feature extraction technique. It highlights regions from where seeded trajectories exhibit more dissimilar behavior. In other word, where small perturbations to the initial seeding value result in large perturbations to the resulting trajectories. This is an important difference and may provide the user with more information about the flow than just observing features alone. Features, by definition, have a distinctive characteristic and frequently, flow trajectories passing through them exhibit dissimilar behavior. Therefore, these regions may be highlighted by the sensitivity field. In addition, regions where the seed of a trajectory does not originate within a feature, but whose trajectory passes through the feature are also highlighted. Therefore, the similarity field may not only highlight features but regions connected by the flow to these features. This may aid the user in understanding the nature of the interesting behavior. Figure 10 (left) shows this benefit. The seeding rake is placed outside of the vortical

behavior of the flow and the streamlines enter this region as they are traced. Rake placement is guided by the sensitivity field.

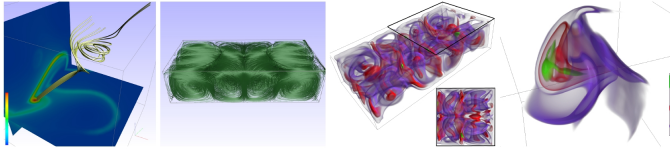### A. Visualization of Streamline Seed Position Sensitivity

Fig. 10: (Left) Visualizing the sensitivity field of Arnold-Beltrami-Childress (ABC) [32] flow using two interactive slice probes. Sensitive seeding regions within the field are visualized directly. Streamlines seeded in red regions are very sensitive to seeding position. Using this approach helps the user locate interesting structures more quickly in the domain than a brute force manual search. In this example, the user quickly navigates the seeding rake to visualize the structure shown using the slice probes for context information. (Middle-Left) Streamlines in the Bernard flow simulation. (Middle-Right) A direct volume rendering of the sensitivity field, $D(p)$, from Bernard flow. As well as highlighting the regions to which streamline seeding is sensitive to change, the DVR provides an overview of interesting seed-based regions within the simulation. (Inset) A top view emphasizing the convection cell structures within the field. (Right) A direct volume rendering of $D(p)$ of the ABC flow and accompanying transfer function. The interesting seeding regions in the flow are highlighted by high sensitivity values mapped to red and green.

Using a slice probe provides a fast and simple tool for direct investigation of streamline seeding position sensitivity. The slice probe may be arbitrarily aligned. Figure 10 (left) depicts two slices of a steady Arnold-Beltrami-Childress (ABC) flow, which describes a closed-form solution of Euler's equation [32]. Color is mapped to the similarity distance $D(p)$. This is true for all color-mapping in this paper unless mentioned otherwise. The color coding conveys positions that are highly sensitive with respect to seeding position, i.e., where a small change in seeding position results in a large change to streamline geometry. The slice probe enables high levels of interaction as it is computationally inexpensive. They are particularly useful when swept through the domain. As in this case, the slice provides context information which may allow the user to navigate to interesting seeding locations more easily.

For DVR we use the Volume Rendering Engine software tool (Voreen) [33]. Figure 10 (middle) shows the corresponding DVR of the left image. Rendering $D(p)$ in this way provides an overview of the seeding sensitivity and its characteristics. Figure 10 (right) shows a direct volume rendering of the sensitivity field of the ABC flow. The green and red regions in the center of the image highlight an interesting vortex structure within the flow. The curvature of streamlines change the most when seeded in or around the vortex core. The transparent regions are less sensitive to streamline seeding position. The geometry of streamlines seeded in these regions changes very little with seed position.
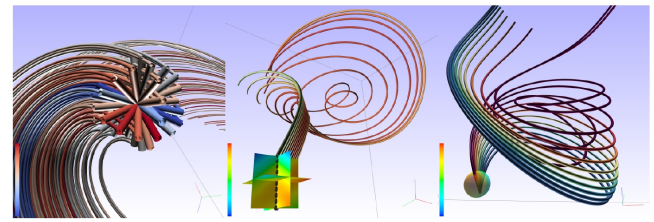

Fig. 11: Previous experiments visualizing seeding rake parameters. (Left) A seeding probe with a dense collection of explicit seeding orientations. This probe produces a cluttered result, especially when streamlines are seeded from all possible positions. It also provides the user with little intuition as to how to orient a rake at the given position. (Middle) A seeding probe with axis-aligned slices. This probe provides insight into the sensitivity information in the vicinity of the seeding rake, through color mapped to sensitivity. However, it suffers from self-occlusion. (Right) A seeding probe depicted as a semitransparent sphere. However, this approach only shows the sensitivity information on the surface - which corresponds to the rake ends. Therefore sensitivity information across the length of the rake is lost. The orientation of the rake can be observed through the use of a transparent surface.

B. Visualization of Seeding Rake Sensitivity

This section introduces techniques used in conjunction with the sensitivity field to provide parameter sensitivity information for a seeding rake. We experimented with a number of different approaches before arriving at the one we found most suitable.

Figure 11 (left) depicts a naive attempt at an interactive probe constructed with a dense collection of rakes at various orientations. Streamlines are seeded from all rakes within the probe. This probe attempts to demonstrate the effect that varying the orientation of the seeding rake has on the resulting streamlines by showing many rakes simultaneously. However, this probe may produce a considerable amount of occlusion as can be seen by the dense rakes and streamlines in the image. It also provides the user with little intuition as to how to orientate a seeding rake at a given position.

The probe shown in Figure 11 (middle) uses three axis aligned slices. The seeding rake is interactively rotated about the center point. This approach reduces occlusion and visual complexity from seeding many streamlines, however, the probe itself is prone to self-occlusion and may occlude part of the seeding rake and resulting streamlines. An ideal tool orientates the rake automatically at an appropriate angle as a function of local flow properties. A somewhat better method utilizes a sphere centered at the seeding rake position. See Figure 11 (right). The sphere provides sensitivity information in all directions around the seeding rake. Occlusion of the seeding rake and the resulting streamlines is reduced through the use of transparency. However, the sphere provides redundant information. Portions of the surface that are aligned with the flow fail to provide further useful information.

The method that we favor is the use of a planar polygon always orthogonal to the flow passing through its center, as in Figure 12. We automatically vary the alignment of the probe

and the seeding rake orthogonal to the local flow field in order to guide the user. There is little benefit from seeding a rake that is tangential to the flow field. In the case of stream surfaces seeded from an interactive rake, a tangential component of the seeding rake to the flow should be avoided [34]. Our approach provides the user with guidance on how to orient their seeding rake as it automatically remains in the plane locally orthogonal to the flow (at its center). Also redundant visualization information is reduced using a local cross-section of the flow. Color is mapped to sensitivity. The seeding probe is depicted as a planar polygon with regular samples distributed across a central axis. The seeding probe sampling can be set to an arbitrary resolution by the user but is set to the same resolution as the data by default. Figure 12 illustrates the effect that a region of high sensitivity can have on streamline geometry with a small change in seeding position. The seeding probe shows a highly sensitive region - the red vertical band on the probe. Here, the user is uncertain as to which direction a given, individual streamline may follow. The streamlines are seeded at equidistant positions along the rake. By comparing neighboring streamlines the effect of a small change in seeding position becomes apparent. Two distinct groups of streamlines can be seen. The sensitive region highlighted by corresponds to the separating region in the flow, where the streamlines flow into either toroid of the Lorenz attractor.
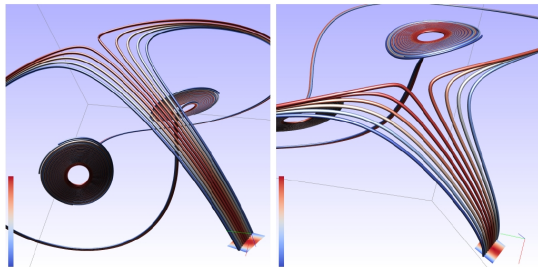

Fig. 12: A set of streamlines depicting a Lorenz attractor. Streamlines are seeded with the seeding rake spanning a sensitive region within the domain. The color-mapping shows streamlines from the center of the rake have the highest sensitivity values. By examining the streamline geometry we observe that a large change results from a small change in seeding position within this region.

V. ADAPTIVE AUTOMATIC INTERSEED SPACING

Our spatial similarity metric allows us to introduce a novel idea of placing seed points in a distribution that produces a more illustrative set of streamlines. Choosing the optimal distance between seed points along a seeding rake is a topic that has received relatively little attention, yet this is a virtually universal parameter for flow visualization using 3D streamlines. In general, the user selects a seeding rake length and the number of streamlines to be seeded from it. Streamline seeds are then distributed at equidistant intervals along the seeding curve. However, this may not be the optimal distribution. Choosing inter-seed distance along a rake is done using trial-and-error in practice. Using our spatial similarity metric it is possible to create an automatic distribution. This is based on the idea of stream surface refinement [35]. In regions of divergence, streamlines are inserted into a stream
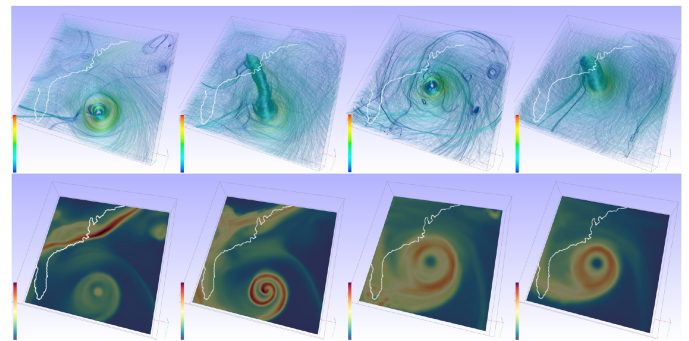

Fig. 13: (Top row) Dense sets of streamlines and pathlines seeded equidistantly for the visualization of Hurricane Isabel. The first two images are taken from t_0 out of 48 time-steps of the simulation. The images are of streamlines and pathlines respectively. The next two images show streamlines and pathlines seeded at t_{24} of the simulation. In both pathline cases the pathlines are traced until $t_{max} = 48$. All integral curves are colored according to local velocity magnitude. (Bottom row) Spatio-temporal sensitivity visualizations of the set of corresponding integral curves in the image above.
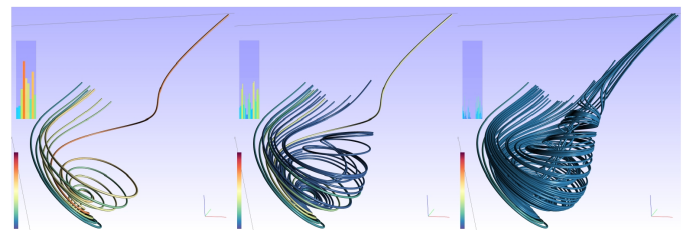

Fig. 14: (Left) Streamlines seeded at equidistant intervals along a seeding rake. The bar graphs shows the individual streamline similarity values. The large values in the bar graph represent a dissimilarity between pairs of neighboring streamlines. (Middle) The previous set of streamline seeds enhanced by our method. New streamlines are automatically seeded along the rake between pairs of streamlines with high dissimilarity. These new streamlines aid the user by visualizing the regions with relatively high streamline variation. A threshold of 50% of the maximum value of $D(p)$ is set. (Right) The resulting streamlines with a smaller threshold of 30% of the maximum value of $D(p)$. Again, this produces a more dense set of streamlines automatically to provide a more detailed visualization. Note that streamlines are only added where needed. If the similarity of neighboring streamlines are within the specified threshold, no insertions take place. This simultaneously helps reduce visual clutter.

surface in order to improve its approximation of the underlying velocity field. Similarly, we introduce extra streamlines along the seeding rake in regions of high dissimilarity between neighbors.

For a given rake, we begin by distributing the seeds uniformly along its length. For each pair of neighboring streamlines, s_p and s_q, we compute $d(s_p, s_q)$. We emphasize this similarity information in the form of a bar graph. The streamline pairs are plotted along the x-axis and their corre-

sponding similarity values are plotted along the y-axis (Figure 14). The range along the y-axis is $[0..D_{max}]$. This enables the user to observe the current streamline sensitivity values in the context of $D(p)$. Next, we allow the user to select a threshold, d_τ, as a percentage of D_{max}. The similarity values of $d(s_p, s_q)$ are compared to d_τ. If $d_i > d_\tau$ a new streamline is introduced between s_p and s_q. Figure 14 demonstrates this. The image on the left shows a set of streamlines distributed evenly across the seeding rake. Note the bar chart visualization shows a large dissimilarity between one pair of streamlines. The next images show seeding distributions enhanced using our method. These streamlines fill the large gaps present using the original seeding pattern.

VI. VISUALIZATION OF TEMPORAL SEEDING POSITION

An extension of our method is used to visualize the seeding parameter sensitivity of unsteady flow fields. In these cases regions in which seeding at different times results in large changes to streamlines or pathlines are highlighted. This has multiple applications. Searching the entire space-time domain for optimal seeding positions can be a daunting task for the user. Therefore, capturing the essential information in a single static image or volume is valuable. Also, this scheme may be used to guide down-sampling a data set temporally. Down-sampling is common due to the large output from modern simulations. It is often desirable to use a down-sampled version of the simulation that fits into main memory for fast performance needed to support interaction. Using our temporal sensitivity field, the user can measure the uncertainty/change between time-steps i.e., $D(p, t_n)$ and $D(p, t_{n+1})$. This can be used as a guide to prevent excessive down-sampling e.g., if the user wants to down-sample the data and encounters a case where $||D(p, t_n) - D(p, t_{n+1})|| > D_\gamma$, the user knows that temporal aliasing may reduce visualization result quality. D_γ is a user defined threshold to control the quality of the down-sampling. In order to handle unsteady flow data, a modification
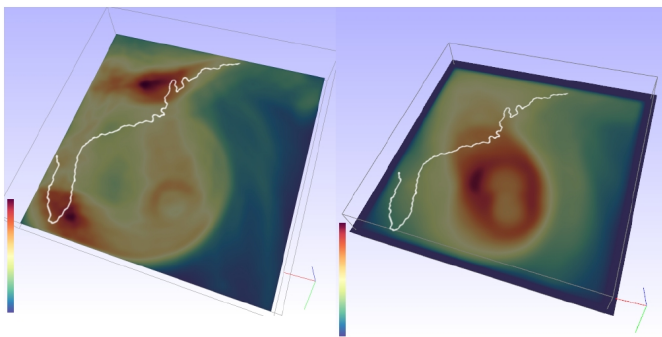

Fig. 15: A slice visualization of $D(p, t)$ from a simulation of Hurricane Isabel. This shows the regions with the most variation to streamline seeding positions over all time steps $0 \leqslant t \leqslant t_{max}$. The path of the hurricane eye and the regions close to it correspond to the most sensitive areas. The left image shows the field constructed using streamlines. The field in the right image was constructed using pathlines seeded at t_0 and traced until $t_{max} = 48$.

is made to how $D(p)$ is computed. Streamlines or pathlines are again seeded at every sample point, $p_i \in \mathbb{R}^3$. However, they

are seeded at every $p_i(t_n)$ for every time-step, t_n. Thus for N time-steps, there are N streamlines or pathlines seeded at p_i. The value of $d(p, t)$ is computed starting with the streamline seeded at time-step t_0 and computing the similarity between $p_i(t_n)$ and $p_i(t_{n+1})$. This is repeated for $t \in T$ with Eq. 11

$$D(p,t) = \sum_{n=0}^{N-2} d(S(p, t_n), S(p, t_{n+1})) \qquad (11)$$

where $S(p_i, t_n)$ is the streamline or pathline seeded at time-step t_n and $S(p_i, t_{n+1})$ is the streamline or pathline seeded at the time-step t_{n+1}.

Figure 15 shows visualizations of the temporal sensitivity field, $D(p, t)$, from a simulation of Hurricane Isabel. This visualization shows that the regions that are most sensitive to seeding over the entire 4D space-time domain. These regions are located around the path that the eye of the hurricane traverses and the area around that path. The color-coding is mapped to $D(p, t)$. Figure 16 is a direct volume rendering of the temporal sensitivity field from a flow behind a square cylinder simulation [36]. The region most sensitive throughout the entire spatio-temporal domain is located behind the square cylinder.

VII. PERFORMANCE

The computation of the spatial and temporal sensitivity fields is a one-time computation. After $D(p, t)$ is derived it can be saved to disk for faster loading. If the user requires fast response times it is possible for the user to take a progressive approach to the sensitivity field construction. Shorter stream-lines can be used to create a less accurate approximation. The streamlines can be lengthened to improve the accuracy. See Figure 17 for an example. In a multi-threaded environment this computation can be performed in the background and the results are refined as they become available. This provides users with immediate feedback and allows them to quickly start exploring the data.

Streamline Length	Computation of $D(p)$ @ (128x32x64)
10	6.79s
50	28.53s
100	55.02s
1000	627.08s

TABLE I: Performance times for computing $D(p)$ using streamlines of varying lengths. These times were measured on the simulation of Bernard flow with domain resolution 128x32x64. The algorithm runs in $O(n)$.

Table 1 shows the computations times for a sample D using varying length streamlines. Streamline integration was performed using a 4th-order Runge-Kutta integrator. The results show the computation time grows linearly with streamline length. We note that the computation does not prohibit visualization of 3D, time-dependent data. Also, the computation time is very fast when compared to a user manually adjusting the input parameters, i.e., searching the entire space-time domain.

A. Domain Expert Review

This work was carried out in close collaboration with a domain expert in fluid mechanics (Professor Harry Yeh). We report his feedback here. The use of sensitivity fields for the visualization of input parameter space in the context of flow visualization is novel. While the initial computation of the sensitivity field may require some seconds or minutes, the benefits of fast navigation through parameter space outweigh the one-time pre-processing cost. Finding expressive seeding positions for streamline rakes can be a laborious and time consuming process. The sensitivity field facilitates this process. It is typically carried out manually by trial and error. For example, the slice visualization of the sensitivity field in Figure 10 (left) aids the navigation of the rake to capture the main vortical structure. This structure is found more quickly than by performing a brute-force search through the domain. And since automatic feature detection algorithms cannot be implemented for every type of feature, general strategies are very helpful.

The use of the seeding probe helps provide confidence to the visualization results. In highly sensitive regions extra care must be taken to ensure that the rake captures the important flow characteristics. As a small change in rake position may make the difference if a certain behavior is captured or not. This leads to the topic of automatic, adaptive inter-seed spacing which is of particular interest. Having the rake automatically distribute the seeds frees the user from this burden. As mentioned previously a small change in position may result in streamlines missing a feature which were present previously. Using the adaptive seeding strategy presents several benefits: (1) The adaptive seeding provides a higher sampling frequency in the required regions and provides a more detailed visualization result ultimately providing greater insight into the seeding behavior and flow structure. (2) Inserting streamlines only where necessary simplifies the visualization by reducing occlusion and visual clutter thus making the results easier to interpret with less visual overload. (3) Adaptive seeding alleviates the problem arising from a feature being lost as the rake is moved slightly. The accompanying video shows an example of an interactive session highlighting this phenomena. This problem could be alleviated by using equidistant seeding and increasing the sampling resolution. However, this places the responsibility of detecting this problem and manually updating the seeding resolution on the user.

The sensitivity field provides additional information about the nature of the flow field which are not communicated using integral curves alone. Section VII-B describes a case study based on the parameter sensitivity data generated from the simulation of Hurricane Isabel.

B. Case Study: Hurricane Isabel

Densely distributed streamlines shown in Figure 13 (top) help observe seeding position sensitivity and the flow pattern of the hurricane. First, the tightly organized flow pattern is shown near the eye of the hurricane. In the area away from the eye, we see the different flow pattern in the upper and lower altitudes: the lower atmosphere tends to spiral into the hurricane, while the flow spreads out in the upper atmosphere. The streamline seeding pattern in the later time $t = t_{24}$ is less coherent than the previous one $t = t_0$; this could be a sign of deterioration as

the hurricane approaches the landmass. Unlike the streamlines, the pathlines maintain the same pattern as those from the first time (top second) including the preferred influence to/from the upper atmosphere in the northeast side of the hurricane. More pathlines are present there.
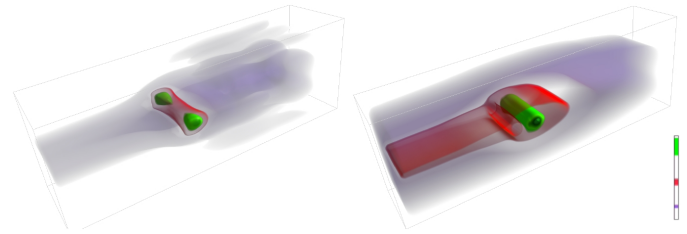

Fig. 16: A DVR visualization of the temporal sensitivity (Eq. 11) field from flow behind a square cylinder [36]. This visualization shows the region behind the square cylinder is the most sensitive over time. The left image shows the spatio-temporal sensitivity field constructed using streamlines. The right image shows the entire spatio-temporal field constructed using pathlines seeded at t_0 and traced until the end of the simulation at $t_{max} = 102$.
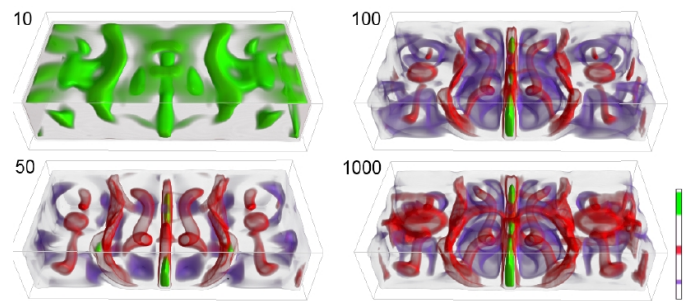

Fig. 17: The iterative derivation of the $D(p)$ of the simulation of Bernard flow. This field is computed by extending the length of the streamlines. The streamline lengths used to create the field are 10, 50, 100 and 1000 points. The color-coding is normalized to streamline length in each image.

As demonstrated, the developed technique is to provide guidelines to determine seeding rake and orientation to construct effective streamline and/or pathline field visualization. Figures 13 (bottom) and 13 however show an additional application. Visualization of the spatio-sensitivity field, $D(p)$, and the spatio-temporal field, $D(p,t)$ in the entire flow domain can be used for the exploratory flow analysis. The high sensitivity means divergence of stream (or path) lines. Figure 13 (bottom left) shows the high sensitivity region of the weather front along the Eastern Seaboard, which must be strengthened by the approaching hurricane. The eye of the hurricane has high sensitivity but that is not the case 24 hours later (see bottom third). This evolution might reflect the growing hurricane at $t = t_0$, and the subsequent weakening at $t = t_{24}$. Also, see the large arc shaped boundary of the high sensitivity ahead of the hurricane. This represents the boundary of air mass influenced by the hurricane; the mass inside of the boundary flows toward the hurricane, while the air mass outside of the boundary flows away from the boundary just like the example of Bernard convection cell shown in Fig. 11 (middle). Later at

time $t = t_{24}$, the appearance of the sensitivity field is different from that of the earlier time. Note that no weather front line can be seen here. It must be pushed further north. It appears that the hurricane has been somewhat diffused and the center of the hurricane (eye) is no longer the location of high sensitivity.

The sensitivity field $D(p,t)$ in terms of pathlines seeded at $t = t_0$ (Fig 13 bottom) also shows the weather front strengthened by the hurricane, but the location is shifted to the north. This is because the front is pushed toward north by the hurricane as it approaches the coast. Just as the sensitivity field of the streamlines, the arc shaped boundary is observed ahead of the hurricane: the boundary separates the air mass from the hurricane side to the outside. The distinct curly shape pattern of the hurricane is intriguing, which implies that the hurricane must form the spiral cell toward the eye. The sensitivity field based on the pathlines at $t = t_{24}$ is grossly different from that of earlier time $t = t_0$. No more spiral pattern is present; instead, a single diffused ring of high sensitivity has emerged. This must be a sign of weakening.

VIII. CONCLUSION

We present methods to quantify and visualize parameters associated with seeding streamlines or pathlines for flow visualization. We define and compare various spatial and temporal similarity metrics which are utilized to derive a sensitivity field. The derived sensitivity field $D(p,t)$ quantifies the predictability of flow trajectories as a function of initial condition, i.e., user-seeding position. This facilitates real-time interaction using our method. Several visualization tools are demonstrated to visualize the sensitivity field in both space and time for seeding rake parameters. We provide a tool highlighting the portions of a seeding rake which benefit from a denser sampling of seeds, producing a more expressive set of streamlines that would not be achieved by naively increasing the sampling frequency. We present an extension to time-dependent data which highlights the most interesting regions over the lifetime of a simulation. We also report the reaction of a fluid dynamics expert in Sections VII-A and VII-B.

Throughout this paper we have focused on presenting the regions that are most sensitive to change. However, the user may also want to see the opposite. Regions where flow is advected in a steady fashion are also interesting to domain users. These regions correspond to the regions of low sensitivity and thus steady flow advection.

For future research we plan to extend this technique to streaklines. One of the central challenges of analyzing this method in conjunction with streaklines is finding the optimal moment to seed integral curves.

REFERENCES

[1] G. Johnson, D. Ebert, C. Hansen, D. Kirk, B. Mark, and H. Pfister. Panel: The Future Visualization Platform. In *Proceedings IEEE Visualization 2004*, pages 569–571, 2004.

[2] R. S. Laramee and R. Kosara. *Human-Centered Visualization Environments*, chapter Future Challenges and Unsolved Problems, pages 231–254. Springer Verlag, 2007. Springer Lecture Notes in Computer Science (LNCS) 4417.

[3] R. H. Abraham and C. D. Shaw. *Dynamics - the Geometry of Behavior*. Addison-Wesley, 1992.

[4] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang. Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13:769–785, Jul/Aug 2007.

[5] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang. Efficient Morse Decompositions of Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):1–15, July/Aug 2008.

[6] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for Visualizing Uncertainty in Vector Fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, September 1996.

[7] S. K. Lodha, A. Pang, R. E. Sheehan, and C. M. Wittenbrink. UFLOW: Visualizing Uncertainty in Fluid Flow. In *Proceedings IEEE Visualization '96*, pages 249–254, October 27–November 1 1996.

[8] S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: Sounding Uncertainty Visualization. In Roni Yagel and Gregory M. Nielson, editors, *Proceedings IEEE Visualization '96*, pages 189–196. IEEE, 1996.

[9] A. Cedilnik and P. Rheingans. Procedural Annotation of Uncertain Information. In *Proceedings IEEE Visualization 2000*, pages 77–84. IEEE Computer Society Technical Committee on Computer Graphics, 2000.

[10] P. J. Rhodes, R. S. Laramee, R. D. Bergeron, and T. M. Sparr. Uncertainty Visualization Methods in Isosurface Rendering. In M. Chover, H. Hagen, and D. Tost, editors, *Eurographics 2003, Short Papers*, pages 83–88. The Eurographics Association, September 1-5 2003.

[11] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to Uncertainty Visualization. 13:370–390, 1996.

[12] V. Verma and A. Pang. Comparative Flow Visualization. *IEEE Transactions on Visualization and Computer Graphics*, 10(6):609–624, 2004.

[13] R. Brown. Animated Visual Vibrations as an Uncertainty Visualization Technique. In *GRAPHITE*, volume 1, pages 84–89. ACM Press, 2004.

[14] Ralf P. Botchen and Daniel Weiskopf. Texture-Based Visualization of Uncertainty in Flow Fields. *Visualization Conference, IEEE*, 0:82, 2005.

[15] J. Sanyal, S. Zhange, J. Dyer, A. Mercer, P. Amburn, and R. Moorhead. Noodles: A Tool for Visualization of Numerical Weather Model Ensemble Uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16:1421–1430, 2010.

[16] J. S. Praßni, T. Ropinski, and K. H. Hinrichs. Uncertainty-Aware Guided Volume Segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1258–1365, 2010.

[17] K. Potter, J. Kniss, R. Riesenfeld, and C. R. Johnson. Visualizing Summary Statistics and Uncertainty. *Computer Graphics Forum*, 29:823–831, 2010.

[18] Ralph Brecheisen, Anna Vilanova, Bram Platel, and Bart ter Haar Romeny. Parameter Sensitivity Visualization for DTI Fiber Tracking. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1441–1448, 2009.

[19] W. Berger, H. Piringer, P. Filzmoser, and E. Gröller. Uncertainty Aware Exploration of Continuous Parameter Spaces Using Multivariate Prediction. *Computer Graphics Forum*, 30(3):911–920, 2011.

[20] M. Hlawitschka, F. Sadlo, and D. Weiskopf. Predictability-Based Adaptive Mouse Interaction for Visual Flow Exploration. In *The Working With Uncertainty Workshop*, Providence, RI, October 2011.

[21] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over Two Decades of Integration-Based, Geometric Flow Visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.

[22] B. Spencer, R. S. Laramee, G. Chen, and E. Zhang. Evenly-Spaced Streamlines for Surfaces: An Image-Based Approach. *Computer Graphics Forum*, 28(6):1618–1631, 2009.

[23] L. Li and H.-W. Shen. Image-Based Streamline Generation and Rendering. *IEEE Transactions on Visualization and Computer Graphics*, 13(3):630–640, 2007.

[24] G. Haller. Distinguished Material Surfaces and Coherent Structures in Three Dimensional Fluid Flows. *Physica D*, 149:248–277, 2001.

[25] Armin Pobitzer, Ronald Peikert, Raphael Fuchs, Benjamin Schindler, Alexander Kuhn, Holger Theisel, Kresimir Matkovic, and Helwig Hauser. On The Way Towards Topology-Based Visualization of Unsteady Flow the State Of The Art. *EuroGraphics 2010 State of the Art Reports*, pages 137–154, 2010.

[26] G. T. Mase. *Continuum Mechanics for Engineers*. CRC Press, 1999.

[27] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch. Feature Extraction and Visualization of Flow Fields. In *Eurographics 2002 State-of-the-Art Reports*, pages 69–100, 2–6 September 2002.

[28] Y. Chen, J. D. Cohen, and J. Krolik. Similarity-Guided Streamline Placement with Error Evaluation. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1448–1455, 2007.

[29] T. Eiter and H. Mannila. Computing Descrete Fréchet Distance. Technical report, Technische Universitt Wien, 1994.

[30] L. Li, H.-S. Hsieh, , and H.-W. Shen. Illustrative Streamline Placement and Visualization. In *IEEE Pacific Visualization Symposium 2008*, pages 79–85. IEEE Computer Society, 2008.

[31] B. Moberts, A. Vilanova, and J. J. van Wijk. Evaluation of Fiber Clustering Methods for Diffusion Tensor Imaging. *IEEE Transaction on Visualization and Computer Graphics*, pages 65–72, 2005.

[32] G. Haller. An Objective Definition of a Vortex. *Journal of Fluid Mechanics*, 525:1–26, 2005.

[33] J. Meyers-Spradow, T. Ropinski, J. Mensmann, and K. H. Hinrichs. Voreen: A Rapid-Prototyping Environment for Ray-Casting-Based Volume Visualizations. *IEEE Computer Graphics and Applications*, 29(6):6–13, 2009.

[34] Ronald Peikert and Filip Sadlo. Topologically Relevant Stream Surfaces for Flow Visualization. In H. Hauser, editor, *Proc. Spring Conference on Computer Graphics*, pages 43–50, April 2009.

[35] J. P. M. Hultquist. Constructing Stream Surfaces in Steady 3D Vector Fields. In *Proceedings IEEE Visualization '92*, pages 171–178, 1992.

[36] S. Camarri, M. V. Salvetti, M. Buffoni, and A. Iollo. Simulation of the Three-Dimensional Flow Around a Square Cylinder Between Parallel Walls at Moderate Reynolds Numbers. In *Congresso di Meccanica ed Applicata*, pages 11–15, 2005.