DOI: 10.1111/cgf.14036
EUROVIS 2020
S. Oeltze-Jafra, N. Smit, and B. Wang
(Guest Editors)

Volume 39 (2020), Number 3
STAR – State of The Art Report

A Survey of Seed Placement and Streamline Selection Techniques

Sudhanshu Sane[1], Roxana Bujack[2], Christoph Garth[3], and Hank Childs[1]

[1]University of Oregon, USA
[2]Los Alamos National Laboratory, USA
[3]University of Kaiserslautern, Germany

Abstract

Streamlines are an extensively utilized flow visualization technique for understanding, verifying, and exploring computational fluid dynamics simulations. One of the major challenges associated with the technique is selecting which streamlines to display. Using a large number of streamlines results in dense, cluttered visualizations, often containing redundant information and occluding important regions, whereas using a small number of streamlines could result in missing key features of the flow. Many solutions to select a representative set of streamlines have been proposed by researchers over the past two decades. In this state-of-the-art report, we analyze and classify seed placement and streamline selection (SPSS) techniques used by the scientific flow visualization community. At a high-level, we classify techniques into automatic and manual techniques, and further divide automatic techniques into three strategies: density-based, feature-based, and similarity-based. Our analysis evaluates the identified strategy groups with respect to focus on regions of interest, minimization of redundancy, and overall computational performance. Finally, we consider the application contexts and tasks for which SPSS techniques are currently applied and have potential applications in the future.

CCS Concepts

• ***Human-centered computing*** → ***Scientific visualization;***

1. Introduction

Flow visualization is a prominent branch of scientific visualization. Its goal is to enable scientists to understand and improve fluid phenomena and computational fluid dynamics models. Further, flow visualization is applicable to many fields, including weather and climate systems, aerodynamics, and turbomachinery design processes. There are many flow visualization techniques, but the most popular is streamlines, the subject of this survey.

The foundational concepts for streamlines are well established. A flow field is typically defined as a vector field over a discretized mesh in the domain. To visualize and encode the behavior of the flow, an integral curve is an effective technique that depicts the path of a particle trajectory in the domain. An integral curve is calculated by first placing a *seed* point in the domain, followed by integrating the path of the trajectory by considering the underlying vector field at each point. In practice, the path of a particle is computed using numerical approximation methods such as Runge Kutta [CK90]. If the vector field is constant over time , i.e., does not evolve, it is considered to be a steady state field and the integral curve traced in such a field is called a *streamline*. If the vector field evolves it is said to be an unsteady state field and an integral curve traced considering this field is called a *pathline*. Given steady state vector field flow visualization is commonly used by scientists for analysis, streamlines are a very popular technique.

Although the computational costs of streamlines can be significant, advances in parallelization have enabled computing large numbers of streamlines efficiently [PPG12,PYK*18]. Using a large number of streamlines, however, does not guarantee a useful visualization. Thus, to assist scientists with the exploration of flow fields in various contexts (planar surface, curved surface, or volume flow), the identification of initial seed placement and the selection of streamlines to visualize has been an active research area. In particular, several research efforts have aimed to automatically generate or select a representative set of streamlines for a given flow field. A survey of these seed placement and streamline selection (SPSS) techniques is the focus of this state-of-the-art report. Our motivation to compile this report is the following:

- Assist application scientists with the selection of SPSS techniques that best meets their needs.
- Provide beginners in flow visualization with an overview of the SPSS research area and an understanding of the theory.
- Inform future flow visualization development efforts.
- Help flow visualization scientists identify research gaps.

Concerning previous work, the survey by McLoughlin et al. [MLP*10] is the most notable study that has addressed seed placement for flow visualization. Although covering the most prominent studies up until that time, SPSS techniques were not a central theme of the study, and the techniques used to identify

a representative set of streamlines have significantly evolved since. Over the past decade, several studies have presented techniques that identify representative streamlines via selection from a random set as opposed to iterative generation methods (typically seen in seed placement algorithms). More recently, the field has adopted machine learning approaches to tackle the challenges associated with identifying representative streamlines. Thus, we believe an up-to-date survey of SPSS techniques for flow visualization would be very helpful to the scientific visualization community. Further, on the topics of general feature extraction or vector field clustering techniques (often employed in SPSS workflows), studies by Post et al. [PVH*03], Laramee et al. [LHD*04, LHZP07], and Pobitzer et al. [PPF*11] provide comprehensive coverage. This report is organized as follows: Section 2 covers background information on SPSS, including the mathematical definition of a streamline, challenges, desired characteristics of visualizations, and evaluation methodology. Sections 3 introduces a high-level classification of techniques and a basic road map for the reader to navigate the survey. Sections 4 and 5 contain details of techniques for each technique and strategy. Section 6 discusses future work, and finally, Section 7 highlights our contributions and concludes the report.

2. Seed Placement and Streamline Selection Background

This section discusses key aspects of seed placement and streamline selection (SPSS) techniques. First, we differentiate between seed placement and streamline selection from a terminology point of view. A **seed placement** algorithm is the process of selecting particle seed locations to calculate useful streamlines. In contrast, a **streamline selection** algorithm is the process of choosing useful streamlines from a large set of precalculated streamlines (typically generated using a random seed placement). In several instances, both seed placement and streamline selection are used together to produce the desired outcome.

The following sections cover the challenges and application contexts, desired characteristics of visualizations, and the evaluation methodologies used with respect to SPSS techniques. Further, we introduce the axes we use in this survey to evaluate different classes of SPSS techniques.

2.1. Streamlines and Pathlines

A streamline is a curve $x_s : \mathbb{R} \to \mathbb{R}^d$ that is everywhere tangential to the instantaneous velocity of an unsteady vector field $v : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ at one fixed time t

$$\frac{dx_s(s)}{ds} = v(x_s(s),t),\ \ x_s(s) = x_s(s_0) + \int_{s_0}^{s} v(x_s(\tau),t)d\tau \qquad (1)$$

while a pathline $x_p : \mathbb{R} \to \mathbb{R}^d$ is tangential to the vector field over time

$$\frac{dx_p(t)}{dt} = v(x_p(t),t),\ \ x_p(t) = x_p(t_0) + \int_{t_0}^{t} v(x_p(\tau),\tau)d\tau. \qquad (2)$$

Both are uniquely defined through a differentiable vector field and the location and time of their seedpoints $x_s(s_0), x_p(t_0)$ [Cod12].

2.2. Challenges

SPSS strategies have been proposed to address various flow visualization tasks. In this report, the majority of research studies propose an algorithm to *generate informative flow visualizations using a representative set of streamlines*. Producing an image using an excessively large number of streamlines can result in a dense and cluttered visualization showing redundant information. However, if the number of streamlines used is too few, important flow features can be missed. Further, in a three-dimensional setting, streamlines often occlude one another. It is desirable that less informative streamlines do not occlude streamlines capturing important features of the flow. Streamline length is an additional consideration when proposing an algorithm. Uniformly placed short streamlines often result in visual artifacts, whereas complete streamlines (i.e., streamlines that only terminate at boundaries or critical points) might result in cluttered, non-uniform distributions.

Although comprising a smaller body of work, we additionally consider SPSS techniques for flow visualization tasks such as:

- Selection of streamlines that are similar to a query streamline, i.e., a user-specified streamline.
- Particle-based flow visualization systems that require seed placement to control particle density distribution.

Our report includes these studies, given the commonality of the underlying objectives and algorithms. These studies are useful in that they address challenges like identifying similar streamlines in an orientation-, position-, and scale-invariant manner, particle density distribution management, and minimization of vector field reconstruction error.

2.3. Desired Characteristics

Verma et al. [VKP00] were the first to explicitly list characteristics that are desired of the selected set of representative streamlines. These characteristics are:

- **Coverage:** Streamlines should not miss interesting regions of the flow field.
- **Uniformity:** Streamlines should be uniformly distributed over the field.
- **Continuity:** From an aesthetic perspective, streamlines should be selected such that they show continuity in the flow, i.e., long streamlines are preferred.

However, these desired characteristics have been modified by researchers and scientists as this area of research has evolved, particularly after considering 3D volume flows, view of the domain, specific features of interest, and information content. Thus, additionally desired characteristics include:

- Visibility of regions of interest (ROI), i.e., occlusion management [MCHM10].
- Smooth transitions or frame coherence when visualizing time-dependent flow or changing viewpoints [GBWT11, MWW*14].
- Retaining spatial perception for depth cues [KFW16].
- Representing maximum information content using the least number of streamlines [LHS08].

Contributions to this area of research have prioritized different

characteristics while advancing or improving on previous work — either from a visualization or computational perspective.

2.4. Evaluation Methodology

The results of SPSS techniques have seen on-going improvements over the past two decades. These improvements have been evaluated qualitatively and/or quantitatively. The majority of these visualization research studies have used a qualitative evaluation to demonstrate that the proposed technique achieves some desired characteristic better than a previous approach. Although fewer in number, multiple studies have quantitatively evaluated a technique by considering the accuracy of vector field reconstruction using the selected streamlines or computational performance.

Qualitative evaluations can be biased based on the specific requirements or objectives of an individual study. Thus, capabilities such as maintaining spatial perception, or highlighting multiple ROI are viewed as "upgrades." To limit the scope of this report, rendering techniques, such as thinning of lines or lighting effects, are not discussed. That being said, choices surrounding the rendering and presentation of streamlines can contribute to and improve our perception and understanding of the flow field.

In this report, we evaluate classes of strategies based on three factors, which enable our analysis of techniques. Further, we incorporate the qualitative and quantitative criteria determined by the studies themselves. These three factors are:

- **Regions of interest (ROI):** Evaluation of whether a technique can identify and focus on ROI.
- **Minimization of redundancy:** Evaluation of whether a technique mitigates the selection of redundant streamlines.
- **Computational performance:** Evaluation of whether a technique can be used in computationally constrained contexts.

The ROI axis is important to evaluate the ability of an algorithm to primarily focus on salient features, avoid occlusion by less important streamlines in 3D, and generate a visualization that naturally draws the user's focus to the ROI of the field [VKP00, MCHM10, GRT13]. The redundancy axis is important to evaluate whether an algorithm is selecting multiple streamlines that convey relatively the same information (e.g., parallel streamlines) [CCK07, LHS08]. Considering redundancy is particularly useful when there is a tradeoff between the number of streamlines that can be used and the total information conveyed by the set of streamlines. The computation axis is useful to understand the cost of a particular strategy and its viability under different scenarios (e.g., interactive, in situ, distributed memory). In total, considering these three axes informs recommendations for which technique to use depending on the application and constraints.

3. Classification

To explore a flow field without any assistance or prior knowledge, a scientist would be required to select locations for seed placement, followed by the generation of streamlines. Based on the computed visualization, the scientist can then iteratively refine their seed placement to produce a meaningful visualization. However, this method can be challenging when dealing with complex flow

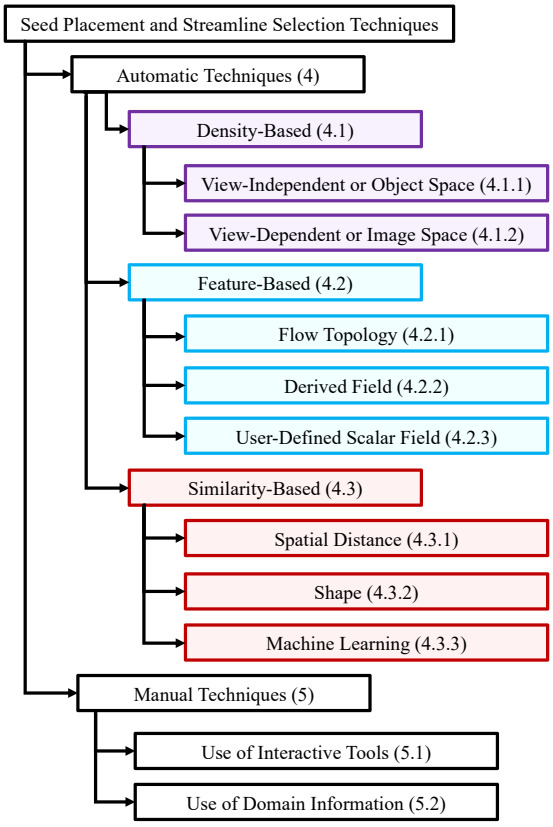

Figure 1: *Classification tree for SPSS techniques. To assist with correlating this classification with Figure 2, we color density-based techniques purple, feature-based techniques blue, and similarity-based techniques red. Finally, each subclass has its corresponding subsection within the survey listed parenthetically.*

fields. To assist scientists with this challenge, two high-level classes of approaches have been developed over the past two decades. The first class of techniques is automatic, i.e., researchers have automated the process by developing algorithms that produce a set of streamlines that convey flow field information. The second class of techniques is manual, i.e., researchers employ interactive methods or leverage domain knowledge to place seed points and manage streamlines. As shown in Figure 1, we further classify automatic techniques into density-based (purple), feature-based (blue), and similarity-based (red) strategies. Within each of these classes, we identify subclasses of techniques. The manual techniques surveyed are classified into interactive tools for the placement or control of streamlines and strategies that use domain-information for seed placement. Given elements of overlap between several SPSS algorithms, we classify studies based on the novelty of the technique contributed to generate or select streamlines.

Automatic techniques have significant diversity in strategy. Figure 2 shows an evaluation of automatic techniques along three axes (introduced in Section 2.4). Given that each class of algorithms contains subclasses and multiple works, our ordering is approximate and relatively general. We base our ordering using comparisons (both qualitative and quantitative) made within research

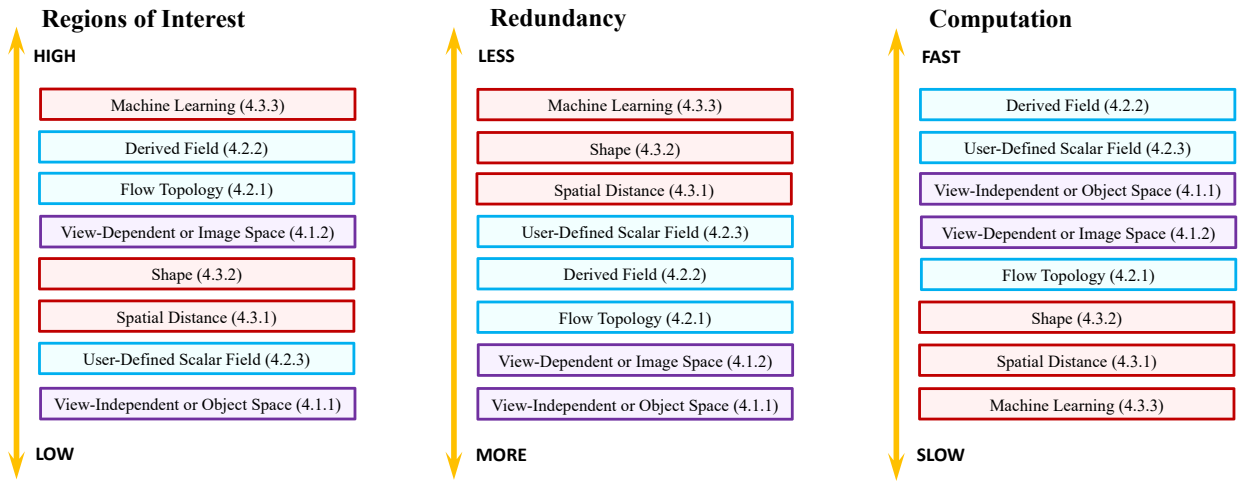

Figure 2: *Approximate, general ordering of the identified categories of seed placement and streamline selection techniques based on three evaluation axes, as discussed in Section 3. We use the same color coding as used in Figure 1. We intend on the rating along the evaluation axes to be used as an approximate guide for the reader to identify categories of interest based on constraints or application contexts.*

works themselves, type of algorithms, scalability of the solution, and our overall understanding of the field. The majority of research studies conduct comparative evaluations with previous work qualitatively and/or quantitatively (e.g., [WLZMI10, KFW16, HTW18] compare against multiple other works). These orders have not been established by conducting new experiments, and certain works within subcategories are exceptions to the position of the entire subcategory along our axis. The objective of the figure is to provide an approximate guide for the reader to navigate the large space of SPSS techniques. Further, the difference between categories might not necessarily be as vast as the position along an axis might suggest. For example, on the ROI axis, although a feature-based strategy will capture ROI better than a view-independent technique and is rated higher, the view-independent technique will indeed have sampled the ROI, albeit without any specific focus.

If the reader desires to find a strategy for their own SPSS problem, this evaluation can be used as an approximate guide. For example, if the reader is interested in a high ability to capture ROI, is indifferent toward redundancy, and desires fast computation, then the reader might be interested in exploring the *Derived Field* class of strategies. Lastly, although we do not organize this survey based on the target contexts of SPSS studies, we believe such a grouping is valuable. Readers can reference Table 3 to identify works that consider a specific target (for example, steady state volume flow).

4. Automatic Techniques

Automatic seed placement or streamline selection algorithms follow a set of rules to generate a distribution/selection of streamlines (or particles in some cases). These algorithms may consider the view direction, properties of the vector field, properties of integrated streamlines and so on. Our classification identifies whether a particular algorithm is primarily contributing a density-based (4.1), feature-based (4.2), or similarity-based (4.3) technique. For example, an algorithm might first extract flow feature locations and strategically place seed points in ROI before placing additional seeds to generate an approximately uniform distribution of streamlines while highlighting flow features — we categorize this as a feature-based technique and not a density-based technique. Often, the motivation of studies will overlap given the desired characteristics are not mutually exclusive, i.e, an algorithm may strive to achieve several desired characteristics in some order of priority.

4.1. Density-Based Techniques

Density-based techniques are typically proposed when a uniform or user-defined distribution of streamlines (not seed points) is the desired outcome of an SPSS algorithm. A uniform distribution of streamlines provides the user with an overview of the entire flow field. Corresponding techniques typically generate approximately evenly-spaced streamlines in object or image space. We categorize the density-based techniques as view-independent (4.1.1) or view-dependent (4.1.2). For view-independent (object space) techniques, the resultant set of streamlines do not change if the view of the domain changes. Whereas, view-dependent (image space) techniques might select different sets of streamlines when the viewpoint of the domain changes.

4.1.1. View-Independent or Object Space Techniques

Density-based view-independent approaches propose algorithms to obtain a uniform or user-defined density distribution of streamlines in object space. The remainder of this section is divided into algorithms that use local or global seeding strategies.

Algorithms Using Local Seeding Strategy

The first use of an automatic seed placement technique to maintain distances between streamlines was by Hultquist et al. [Hul92]. Hultquist et al.'s early work considered seed point addition and removal in the context of stream surface construction. After seed points are initialized as a rake, the distance between particles is tracked as particle trajectories (streamlines) are integrated. Based on the premise that to achieve a good surface visualization an approximately uniform spacing between streamlines is desired, new

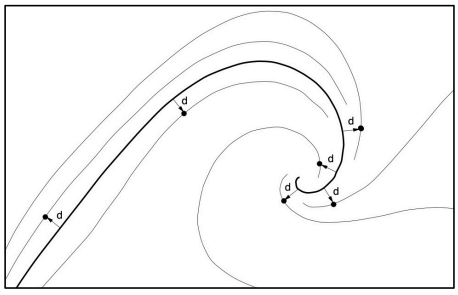

Figure 3: *Candidate seed points are identified at locations a minimum separating distance away from the initial streamline (thick) [JL97]. Image courtesy of Jobard et al.*

seeds are added or existing seeds are merged based on a user-defined neighboring particle distance criterion.

Max et al. [MCG94] used evenly-spaced short streamlines to visualize a 3D vector field on a contour surface. They considered several projections to visualize the streamlines. While they evaluated different projections (Eye, Normal, XY, and Cylinder) on the 3D surface and transitions of those projections as the view changes, a precomputation phase involved seed placement and particle tracing in object space. To allow streamlines to be traced for long distances before they get too close to each other, the initial positions of seed points are chosen on an integer lattice in a spatially hierarchical manner. A streamline length threshold is used to determine the minimal length of accepted streamlines. A streamline grows until it reaches a surface edge, a singularity in the field, or becomes too close to another streamline.

Jobard and Lefer extended the work done by Max et al. [MCG94] and proposed an effective and popular single pass method for placement of long evenly-spaced streamlines in a 2D steady state field [JL97]. The method can achieve visualizations ranging from dense texture-like to sparse hand-drawing styles by only setting the separating distance, denoted by d_{sep}, between adjacent streamlines. The algorithm initially places a random seed point and integrates a new streamline backward and forward until some termination criterion is met. The first streamline is used to calculate a set of candidate seed points d_{sep} distance away from the streamline. The candidate seed points are added to a queue to be evaluated. Each candidate seed point is used as a starting location to integrate a streamline until it is within some distance d_{test} (a fraction of d_{sep}) from existing streamlines. Figure 3 illustrates seed points, a user-defined distance away from an initial streamline, used to integrate new streamlines. If the integrated streamline is accepted, then the new streamline contributes a set of candidate seed points to the existing queue. To accelerate the computation process, Jobard and Lefer proposed two optimizations:

- Streamlines consist of a set of evenly-spaced sample points that are a distance smaller than d_{sep} apart. Only these sample points are considered in distance computations.
- A cartesian grid with cell side d_{sep} is superposed on the domain to support the binning of sample points and limit distance computations to surrounding cells.

These optimizations have been employed in several following research works. The algorithm achieved placement quality as good as previous techniques, i.e., work by Turk and Banks [TB96] (described in Section 4.1.2), while significantly improving computation speeds. Jobard and Lefer extended their initial work to propose a multiresolution technique for steady state flow [JL01] and an approach for creating animations to visualize unsteady flow [JL00]. To generate a sequence of streamline-based images of a vector field with different density (multiresolution), they computed an initial set of streamlines for a large separating distance value. The resulting streamlines then form an initial set of streamlines for the next level, i.e., an image that has a higher streamline density and uses a smaller separating distance value. This process is performed for the desired number of levels of streamline density. The shortcoming of this approach is that streamlines traced for later levels were shorter due to the existence of the initial set of longer streamlines.

For the visualization of unsteady flow in 2D, they proposed a feed-forward algorithm that used *reference* streamlines from one time step to select *corresponding* streamlines in the next time step. Sample points of *reference* streamlines act as initial seed locations to generate candidate streamlines. The best candidate streamline, based on an L2-norm correlation measure, is selected as a corresponding streamline in the next time step. If required, additional streamlines are calculated to obtain a uniform distribution. By correlating instantaneous visualizations of the vector field at the streamline level, they animated 2D unsteady flow visualization.

Mattausch et al. [MTHG03] adopted Jobard's and Lefer's algorithm [JL97] with the aim to improve focus+context techniques and the spatial perception of 3D flow fields. To extend to 3D, six candidate seed points are calculated at a distance d_{sep} for every sample point on a streamline. Additionally, they improved the multiresolution technique presented by Jobard and Lefer [JL01] by preventing the generation of shorter streamlines for higher levels of detail.

Jobard's and Lefer's algorithm has been utilized as an intermediate step for texture-based flow visualization techniques and domains such as DTI Fiber Tracking [VBVP04, MSE*05]. Li et al. [GBH03] presented Chameleon, a texture-based rendering framework, which decouples the calculation of streamlines and the mapping of visual attributes, allowing flexible control of the visual appearance of the vector field. The seed placement algorithm is employed to control the length and density of the generated streamlines. A trace volume is created using a dense set of evenly-spaced streamlines and their geometric properties. The trace volume can then be combined with varying input appearance textures to produce a wide range of effects interactively. Shen et al. [SBL04] extended the Chameleon framework to support unsteady flow fields by calculating pathlines instead of streamlines. The algorithm tracked pathline segment intersections and trace volume updates during rendering. However, the study did not address the distribution of pathlines across the domain over time.

Employing a pipeline similar to Chameleon, Helgeland et al. [HE06] proposed a method to use evenly-distributed particles as input for a texture-based visualization of unsteady flow in 3D. The algorithm outputs a point set, i.e., seed locations, instead of a set of streamlines. Using an initially random pool of seed points, Jobard's and Lefer's algorithm [JL97] is applied to identify the subset of seed points that generate a set of streamlines d_{sep} distance apart. The resultant point set is used to generate streamlines us-

ing a texture-based method (for example, Seed LIC [HA04]) for each time step. After each advection step, cluttering is avoided by removing particles that are less than d_{test} distance apart. While particles leaving the domain are naturally removed, particles are added to account for inflow. A seed point is added to the center of boundary voxels if a fixed length streamline traced from it is d_{sep} distance from existing streamlines. Overall, particle density is maintained by injecting particles into areas with low density without exceeding a user-defined maximum number of seed points for the domain.

Algorithms Using Global Seeding Strategy

Unlike local seeding strategies that place seeds in the vicinity of previously placed streamlines, Mebarki et al. [MAD05] proposed to place seeds furthest away from all previously placed streamlines. Using an approach proposed by Chew et al. [Che93] that had already been successfully applied to point sampling and mesh refinement [ELPZ97, EG01, OB03], Mebarki et al. place new seed points at the center of the biggest voids within the domain. Using Delaunay triangulation to identify voids in the domain, the circumcenter of the triangle with the largest circumradius is chosen as the next seed location. Streamlines, approximated using a set of sample points, are inserted one at a time and are traced until a minimum separating distance criterion is violated. Processing a priority queue of triangles, sorted by circumradius and with circumcircle diameter larger than the separating distance, the algorithm ends when the priority queue is empty. The computation of the process is significantly optimized by only using every n^{th} sample point to calculate the Delaunay triangulation and only adding triangles incident to the streamline end points to the priority queue. Placing seed points farthest away from existing streamlines results in long streamlines, improving on the quality of representative streamlines by reducing streamline discontinuities. Mebarki et al. demonstrated reduced execution time compared to Jobard and Lefer [JL97] for 2D domains, while retaining the placement quality of Turk and Banks [TB96].

To study flow phenomena near-wall regions or boundaries, i.e., curved surfaces in 3D, Rosanwo et al. [RPP*09] proposed a greedy seed placement algorithm. Similar to previous approaches, a single distance δ is used to control streamline density. However, the method avoids the computation of geodesic distances and reduces the search space for seed placement to a set of curves. The algorithm employs two sets of streamlines, namely, *primal* and *dual* streamlines. Primal streamlines are tangential to the vector field at every point and are used to visualize flow phenomena. Used to approximate the largest uncovered areas in the domain, dual streamlines are a supplementary set of streamlines that are orthogonal to the vector field at every point. A small set of both primal and dual streamlines can be initialized either randomly or by using flow field topology. Given the orthogonal directions of the two sets of streamlines, they intersect at several locations. Segments of primal streamlines are stored in a priority queue P, ordered by arc length. Similarly, segments of dual streamlines are stored in a priority queue D, ordered by arc length. The algorithm iteratively selects the longest arc in P or D and places a seed at the midpoint to calculate the next streamline, followed by both queues being updated based on new intersections and segments. The algorithm stops when the length of the longest segment is less than twice the value of δ. Figure 4 illustrates the steps involved in the algorithm. An informed placement of the initial set of streamlines can reduce time to convergence for the algorithm and highlight flow topology resulting in speedups of 2x-3x and improved streamline placement quality over previous approaches [JL97, MAD05, TB96] when evaluating streamline placement for planar surfaces.

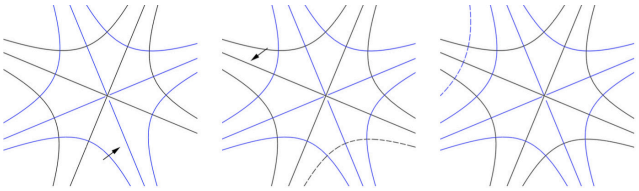

Figure 4: *The dual streamlines algorithm proceeds by identifying the largest segment in two sets (black and blue, i.e., primal and dual streamlines respectively) of streamline segments [RPP*09]. Left: the arrow indicates the midpoint of the largest segment, i.e., the new seed location to calculate a streamline. Middle: the arrow indicates the next largest segment, i.e., next streamline seed location. Image contains streamline generated from seed point in left image (dotted-line). Right: Result after placement of next streamline (dotted-line). Image courtesy of Rosanwo et al.*

Zhang et al. [ZWZ*13, ZSW10] proposed a method to place streamlines in parallel for 2D flow fields. They define local tracing areas (LTAs) as subdomains enclosed by streamlines and/or field borders, i.e., regions where the tracing of streamlines is localized. Using an irregular domain decomposition strategy, the initial LTA is recursively partitioned into hierarchical LTAs. Within an LTA, if a valid seeding area (VSA, determined by streamline proximity criteria) exists, a new seed point is placed at the centroid of the biggest VSA. They use a cell marking technique, instead of performing distance checking, to mark zones where seeds can be placed and streamlines traced. The authors further extended the algorithm to support multiresolution and 3D flow fields [ZZS11, ZNZ*14]. A comparison with Mebarki et al. [MAD05] showed equivalent or better placement quality and an order of magnitude faster computation on parallel hardware.

Analysis: View-independent algorithms inherently provide coverage of the entire domain, i.e., object space, either by generating nearby candidate seeds from existing streamlines or by placing seeds in the largest voids in the domain. However, given the primary objective is to achieve a desired density of streamlines, none of these algorithms have a focused ability to capture ROI. Instead, these algorithms primarily operate on the concept of maintaining a minimum separating distance. As a consequence, generated streamlines can be similar to existing streamlines since there is no measure to account for redundancy. Concerning computation, the majority of these algorithms adopt an iterative serial algorithm. Tracing streamlines one at a time would limit applicability when considering large vector field data sets across multiple nodes or under in situ constraints.

View-independent algorithms support fast exploration once a set of streamlines is generated given only a single set of streamlines serves all viewpoints. Further, these techniques are typically limited to planar or curved surface flows or are used in conjunction with other techniques such as texture-based flow visualization or interactive methods to address occlusion in volume flows.

4.1.2. View-Dependent or Image Space Techniques

By only considering object-space, view-independent techniques did not address occlusion problems that pose a significant challenge when exploring volume flows. View-dependent techniques presented in this section, take the image viewed by the user into account or use the current image as a guide to determine the placement of seed points and the selection of streamlines. The remainder of this section is divided based on whether algorithms use filters, image space seeding, or occlusion and projection of streamlines.

Algorithms Using Filters

Pioneering work in the field of streamline placement, Turk and Banks [TB96] proposed the use of a stochastic mechanism to iteratively refine the placement of streamlines to visualize 2D steady state flow. The approach is based on the idea that for a given image containing a set of streamlines, the application of a low-pass filter to its corresponding binary image should result in an evenly-gray image if the streamlines are uniformly distributed. Areas with streamlines cluttered will have bright pixel values while sparsely represented areas will remain dark in the low-pass filtered image. The energy of the streamline image can be quantified as the sum of difference with a given gray-scale value at each pixel of the low-pass filtered image. The density of streamlines can be controlled by adjusting the size of low-pass filters and optimization of the streamline distribution is realized via iteratively minimizing the energy function. For this work, the filter applied is a circularly symmetric filter kernel from a basis function of cubic Hermite interpolation.

Beginning with streamlines generated from seed points at vertices of a 2D grid, where each streamline has an associated energy contribution, the set of streamlines is modified until the desired energy threshold is reached. The algorithm considers moving, lengthening, shortening, deleting, inserting, and combining streamlines based on energy. The streamline modifications are either proposed by an oracle (50%) or are random (50%) to prevent any oracle bias. The oracle speeds up the convergence of the optimization by a factor of 3x-5x. To propose effective changes, the oracle uses image information to identify sparse regions and maintains a priority queue of streamlines based on an energy level. Thus, the oracle suggests regions to insert streamlines or how to reduce the energy contribution of the most energetic streamlines. If the modification lowers the overall energy value of the image, the change is accepted, otherwise, the change is rejected. The process continues until the energy function reaches a threshold or the accepted changes are rare. Although this approach produced high quality streamline placement, it is computationally expensive given long convergence times.

Mao et al. [MHHI98] extended the Turk and Banks algorithm to uniformly distribute streamlines on a curvilinear grid. They use the image-guided algorithm because density distribution on curvilinear grids, which are anisometric, is hard to achieve with distance-based approaches. First, a mapping of vectors on the curvilinear surface to computational space is performed. To account for the mapping distortion caused by an uneven grid density on a curvilinear grid, a new energy function is employed. Using a Poisson ellipse sampling to distribute a set of rectangular windows in computational space, the streamline density is locally adapted to the inverse of the grid density in physical space. The use of such an energy function ensures the generated set of streamlines is evenly distributed after being mapped back onto the 3D surface.

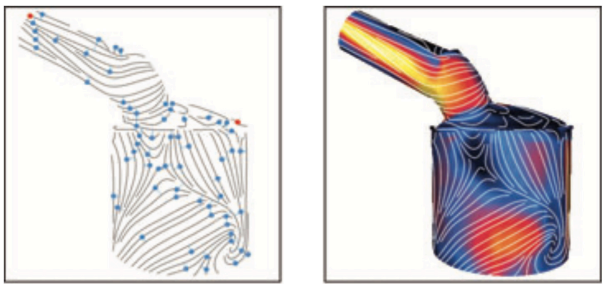

Figure 5: *Placement of streamlines on surfaces in 3D flow using the algorithm by Spencer et al. [SLCZ09]. Left: The image shows the seed locations in image space. Grid-based seeds are shown in red and vector field-based seeds are shown in blue. Right: Final streamlines placement. Image courtesy of Spencer et al.*

Algorithms Using Image Space Seeding

Uniformly distributed streamlines in 3D space are not guaranteed to be evenly spaced in their 2D projection. To avoid clutter in a 3D streamline visualization, Li et al. [LS07] performed seed placement and streamline termination in image space, and streamline advection in object space. The algorithm operated similarly to Jobard's and Lefer's algorithm [JL97], except that candidate seed points for a streamline were d_{sep} apart from the streamline in image space. Thus, even though a 3D volume flow is under consideration, for every sample point of the streamline only two possible candidate points are identified. A streamline is advected in object space and terminated if it is within d_{sep} from another streamline in image space. Further, a streamline closer to the viewpoint is preferred to another far behind. To support importance-driven seed placement, their algorithm decoupled seed point generation and streamline spacing control. A set of seed points is produced using a process that stochastically generates more seeds in ROI, followed by tracing the corresponding streamlines in object space. To avoid clutter, streamlines that violate spacing requirements in image space are deleted. This approach by Li et al. was the first work that used an image space-based seeding strategy.

Spencer et al. [SLCZ09] presented an evenly-spaced streamline seeding algorithm for vector fields defined on surfaces in 3D space. The algorithm is capable of generating both sparse and dense representations of the flow and can handle large, complex, unstructured grids with holes and discontinuities. Streamlines are only integrated for the portions of the surface visible in image space. The advection strategy removes the need to perform streamline tracing on a triangular mesh and instead projects the vector field onto the image plane. Seed placement and streamline integration are then performed in image space. The flow data is stored in a "velocity" image where each pixel stores the flow velocity on the surface and a 16-bit representation of the z-buffer representing the distance of the surface. The use of a z-buffer allows the algorithm to disregard non-visible portions of the surface and plays an important role in detecting discontinuities or edges. The algorithm places seed points, called grid-based seeds, in every cell of the mesh with non-zero depth. Next, it generates vector field-based seeds, i.e., candi-

date seed points, in a manner similar to Jobard's and Lefer's algorithm. They terminate a streamline when the proximity to another streamline drops below d_{test} or when z-buffer drops to zero or the change in z-buffer exceeds a user-defined threshold. Using both sets of seeds in combination ensures all visible sections (there are potential geometric discontinuities arising from edges and occluding surfaces) have a uniform distribution of streamlines. To avoid terminating streamlines near edges due to proximity in image space (greater distance apart in object space) they check for approximately the same z-buffer value. To improve depth perception in the visualization, the value of d_{sep} varies with depth. The idea of reducing any complex surface to a 2D problem results in a computationally efficient algorithm. Figure 5 illustrates a result of the approach. Spencer et al. used a GPU to improve rendering times and showed their streamline generation is faster than an implementation of Jobard's and Lefer's algorithm in 3D object space.

Algorithms Using Occlusion and Projection

Based on the use of contours to visualize scalar fields, Annen et al. [ATR*08] introduced the concept of vector field contours for flow exploration. The proposed algorithm generates isolated streamline which displays behavior similar to that of isocontours. The approach is view-dependent as seeding structures are identified by locating points where the dot product of the view direction and the vector field is zero, and a seed that takes one infinitesimal integration step preserves that condition. Multiple rendering passes are applied to extract the seeding structure with curvature being used in a similar manner as an isovalue in a scalar field. Streamlines are then integrated forward and backward until the dot product of the vector at the streamline position and the view direction exceeds a threshold. The extraction and rendering of the vector field contours is inter-frame coherent providing interactive exploration.

Marchesin et al. [MCHM10] selected streamlines that contribute to understanding flow field characteristics, while simultaneously accounting for clutter in a given view. The approach uses streamline features and the occlusion caused by it to decide whether to include a particular streamline. The four-stage algorithm begins with the computation of a random pool of streamlines. Projecting all the computed streamlines onto an occupancy buffer helps identify highly occluded regions for a given view. Given the importance of swirling lines to understand flow behavior, the occupancy buffer does not account for self-occlusion caused by a single streamline and simply measures the screen space footprint. Next, for each pixel, the number of streamlines projecting onto this pixel is calculated. The third stage, a pruning step, evaluates information conveyed and occlusion caused by a streamline. To determine the quantity of information conveyed by the streamline, the linear and angular entropy values of segments of a streamline, i.e., the local length and angular variation, are used. Additionally, they consider an overlap value to determine the occlusion caused for a given view. Combining these values, they present a streamline metric which is a weighted sum of the linear and angular entropies divided by the average overlap. Sorting streamlines by their score, the streamlines with the lowest score are iteratively removed, followed by an update of the occupancy buffer, and affected streamlines. The final stage of the algorithm decomposes the occupancy buffer into a number of tiles and computes the average occupancy for each tile.

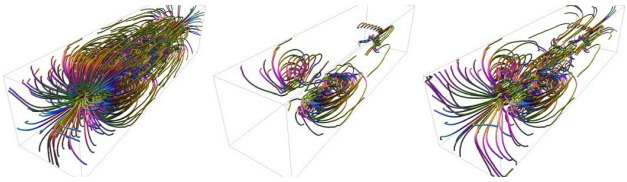

Figure 6: *Stages of the Marchesin et al. [MCHM10] algorithm. Left: an initial dense pool of streamlines. Middle: streamlines with high importance metric rating. Right: Final result after the addition of streamlines to create a more uniform distribution of streamlines in image space. Image courtesy of Marchesin et al.*

Seeding a small pool of random streamlines from the tile with the lowest occupancy, the streamline resulting in the least occlusion is retained. This process is repeated until all tiles have a non-zero occupancy. Figure 6 illustrates a result of the approach and shows the evolution of the visualization over the algorithm stages. The approach captured features of the flow better than previous view-dependent methods and required a GPU for fast computation.

Günther et al. [GBWT11] presented an interactive, view-dependent, and inter-frame coherent flow visualization technique whose results are dependent on user-driven seed placement. The method has an initial preprocessing step, which involves both user-guided seed placement using a seed box, and random placement to generate streamlines that cover the entire flow field. For each streamline, a *screen contribution* value is computed by using a cubic Hermite interpolation function to map the number of visible pixels of the streamline to a transparency value. The screen contribution values are used to determine which streamlines are visible to the user for a given view and fade out streamlines with only minor contributions. Given one important region of the flow can occlude another, the user can selectively place seed boxes in order to focus on certain regions. To support exploring regions of coherent flow, the user can highlight a set of similar streamlines by selecting a single streamline. Streamlines in a limited screen-space neighborhood window of the selected streamline are evaluated for similarity using linear and angular entropy.

Günther et al. [GRT13] extended their previous work [GBWT11] by adopting a global line selection strategy. Starting with an initially dense domain sampling, the algorithm computes the opacity for every streamline segment in the field as a solution to a bounded-variable least-squares optimization problem. Metrics such as curvature, linear entropy, angular entropy, scalar entropy, segment length, or *screen contribution* can be used as an importance measure of a streamline segment. Depending on the metrics chosen the algorithm highlights relevant features in the flow field by minimizing the occlusion caused by other streamlines. While the optimization problem is based on the total number of streamline segments in the flow, the number of segments increases significantly and can become a bottleneck when considering unsteady state flow. To tackle the challenge of 3D unsteady flow, Günther et al. [GRT14] modify their approach and employ a hierarchical representation of an integral curve and consider only a view-dependent set of candidate segments for the optimization process. Günther et al. use the GPU to achieve frame coherent, time coherent, and interactive flow exploration, thus improving on previous research. Figure 7 shows a sample result of the algorithm.

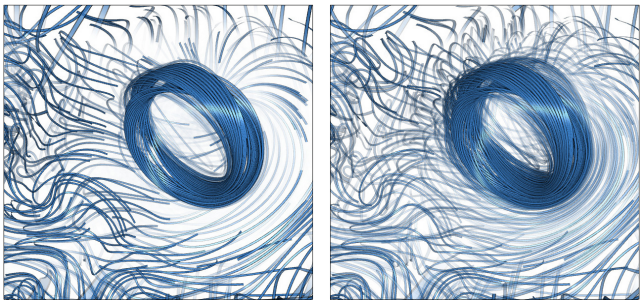

Figure 7: *A comparison of opacity optimization (left) and hierarchical approach (right) demonstrated by the algorithm by Günther et al. [GRT14]. Image courtesy of Günther et al.*

Ma et al. [MWS13] presented a view-dependent streamline selection algorithm that evaluates the information content of streamlines. As a preprocessing step, a dense set of streamlines intersecting every voxel in the domain is computed. Next, for every sample viewpoint, the streamlines are sorted on the basis of importance. The streamline importance measure consists of entropy (considering both direction and magnitude) measured along the streamline, an evaluation of how much entropy is preserved for a given 2D projection, and a shape characteristic metric indicating whether the streamlines characteristics are being conveyed for a given viewpoint. The last two factors together form a view-dependent importance measure for each streamline for each viewpoint. First, a set of view-independent representative streamlines are identified by inserting streamlines into a priority queue based on the summation of their view-dependent importance measure for every view. A minimum threshold distance is used to avoid selecting redundant streamlines. To generate the view-dependent set of streamlines, the top-ranked streamlines for a viewpoint are combined with the highest rated streamlines from the view-independent set. Further, to maintain coherence as the viewpoint is changed, streamlines from a previous viewpoint are retained. A density map is employed to determine uncovered regions before rendering the final visualization. Their algorithm was able to generate fewer redundant streamlines compared to Marchesin et al. [MCHM10].

Analysis: View-dependent or image space algorithms have been proposed to account for occlusion that arises from visualizing streamlines in 3D. With respect to ROI, these techniques have evolved from initially only considering uniform placement on planar or curved surfaces, to limiting streamline calculation to the image space, to evaluating occlusion, projection, and information conveyed by a streamline before selection. Thus, the current state of the art includes view-dependent algorithms that are capable of highlighting ROI for a given viewing angle. However, in general, these methods do not identify or strategically place seeds closer to ROI. Additionally, an important feature can occlude a second important feature resulting in only the streamlines in the foreground being selected. Most view-dependent algorithms do not consider the similarity between streamlines selected and thus, selections can be redundant if pruning steps are not performed. Early methods using filters were iterative and required long convergence times, whereas more recent algorithms are faster and enable interactive exploration of the flow field. However, during exploration, streamlines need to be reselected for every change in viewing angle and frame coherence techniques need to be adopted. View-dependent techniques would be suitable when considering automated in situ flow visualization or in an interactive setting with a user responsible for selecting the appropriate viewing angle.

4.2. Feature-Based Techniques

Feature-based techniques make use of available vector field information to guide the seed placement or streamline selection. Prioritizing coverage of ROI of the flow field over a uniform distribution, these approaches aim to first take measures to ensure seed placement occurs near salient flow features (e.g., critical points shown in Figure 8). We classify feature-based techniques depending on whether they explicitly extract flow field topology (4.2.1) for precise information or allow derived (4.2.2) and user-defined (4.2.3) scalar field to guide the algorithm. Approaches use a derived scalar field in order to either capture some particular flow field behavior or as an alternative approach to capture salient flow features without explicitly calculating their locations. In addition to strategies to highlight features, feature-based techniques often utilize density-based strategies to calculate streamlines in less interesting regions.

4.2.1. Flow Topology Techniques

Flow topology-based techniques calculate the locations of critical points or the separatrices and then use the information for seed placement. The remainder of this section is divided based on whether an algorithm uses critical point locations or directly uses separatrices as an initial set of representation streamlines.

Algorithms Using Critical Point Locations

To visualize nested weather models, Treinish [Tre00] proposed to use a combination of critical point analysis and a filter similar to Turk and Banks [TB96]. Deriving a set of seeds using a low-order approximate critical point analysis, an initial set of streamlines is computed. A low-bandpass filter is subsequently applied to the entire forecasted velocity field, as opposed to an image of the streamlines, to identify regions with a relatively large change in wind speeds. Seed points are placed in these regions to calculate additional streamlines. The technique was superior to using uniformly sampled seed points and captured detailed features from the forecast. However, this particular work did not provide seed placement specifics in relation to critical points or ROI.

Verma et al. [VKP00] proposed the use of critical point-specific

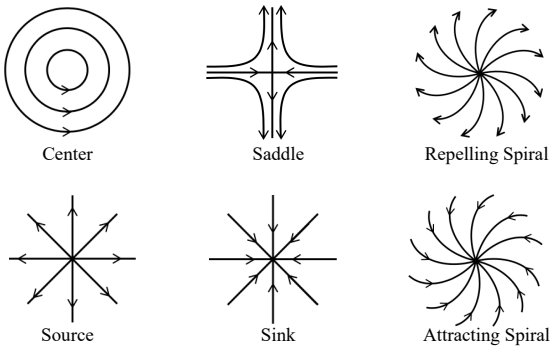

Figure 8: *Types of critical points in 2D flows.*

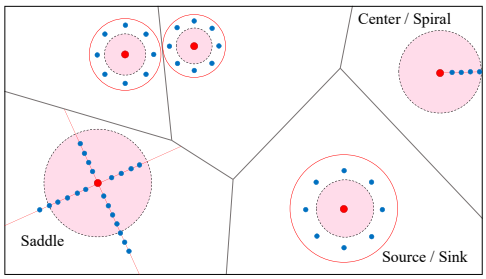

Figure 9: *Notional example showing use of seed templates for various critical points (red) and regions of influence (pink circle) identified by the flow-guided algorithm. Image inspired by [VKP00].*

seeding templates to capture flow behavior in the vicinity of critical points. The algorithm first identifies the locations and types of critical points in the 2D flow field and then segments the domain into approximate critical point neighborhoods. For the approximation, a Voronoi diagram partitions the flow into regions containing similar flow behavior. A second objective of the approach is to provide sufficient coverage of non-critical regions. After tracing long streamlines using the template seeds, a region of influence is determined for each critical point. In the spaces outside the regions of influence, Poisson disk distribution [Coo86] is used to place additional seed points, with streamlines generation following Jobard's and Lefer's algorithm [JL97]. Figure 9 shows a sample field with critical points, corresponding templates, and regions of influence, and the field partitioning such that each partition contains a single critical point. The algorithm was able to better capture flow behavior around critical points in both dense and sparse flow representations when compared to previous techniques [TB96] while being computationally faster when selecting a greater number of streamlines.

In addition to extending the template-based approach to 3D steady state flow, Ye et al. [YKP05] proposed improvements to the Verma et al. algorithm [VKP00]. To account for the distance between and relative strength of critical points, they change the shape of the templates by mapping how eigenvalues of one critical point evolve into the eigenvalues of another. To determine the size of seed templates, separating regions are calculated. To calculate separating regions in 3D for each critical point, instead of using expensive full topological analyses [TWHS03, MBS*04], an approximation is used. The size of the seeding template is set to a quarter of the distance to the nearest-other critical point. Poisson sphere distribution is used to fill areas between critical point regions of influence. The algorithm involves a streamline filtering step to provide a less cluttered visualization. Streamlines are filtered on the basis of length, accumulated winding angle, and proximity to other streamlines. First removing short streamlines with low winding angles, followed by identifying a single representative streamline for a set of streamlines that have a similar start, end, and centroid location. Finally, for a cell with a high streamline count, denoting a dense region, streamlines with high winding angles are removed.

Liu et al. [LMG06] proposed an evenly-spaced streamline algorithm (ADVESS) that employs two queues of candidate seeds. The primary queue consists of an initial set of seeds generated using the templates used by Verma et al. [VKP00]. Candidate seeds generated from the initial location of the streamline seeds are also added to the primary queue to maximize the effect of the seeding patterns. The secondary queue, used only if the primary queue is empty, consists of candidate seeds generated along the calculated streamlines. As an optimization, cubic Hermite polynomial interpolation using large sample spacing is used to represent a streamline using fewer points. As a consequence, the cost of distance checking is reduced due to fewer sample points considered. The algorithm terminates when all the seeds are processed, i.e., the queues are empty.

An additional improvement on previous evenly-spaced streamline algorithms is the use of an adaptive d_{test} value based on the local variance measured at each grid point in the 2D field. Appropriately scaling the value of d_{test} causes fewer cavities in the streamline placement. Further, the authors propose a robust loop detection technique that limits a streamline loop to a single cycle [LMI07]. Employing the algorithm as one part of a hybrid seed placement approach, Liu et al. [LM08] presented a view-dependent approach for seed placement on a planar or curved surface. Using the double queue strategy differently, Poisson disk distribution is used to push a set of seeds to the secondary queue and begin the process. Candidate seeds introduced by the seed location of the accepted streamline are stored in the primary queue, and other candidate seeds are stored in the secondary queue. The approach is used for the purposes of image space seed placement to fill spaces after a primary set of physical space seeds are used to generate streamlines that are reused and lengthened between view frames. The combination of the two strategies provides a temporally coherent visualization. In comparison to previous approaches, the algorithm achieved placement quality better than Jobard and Lefer [JL97] and as good as Mebarki et al. [MAD05] with loop detection, in addition to being computationally faster than both.

Ding et al. [DZC*12] present a technique to maintain temporal coherence when viewing unsteady 2D flow fields by using a moving mesh method. Another extension of Jobard's and Lefer's algorithm, they first extract critical points to calculate an initial set of candidate seeds. Using Poisson disk distribution, they place seed points in ROI and add them to the queue of candidate seeds. They move the seeds towards the critical features by creating and deforming an auxiliary mesh along with the evolution of the vector field. They use a similar feed-forward pipeline system to identify corresponding streamlines to maintain temporal coherence across frames.

Algorithms Using Separatrices As Initial Set

Chen et al. [CML*07] modified Jobard's and Lefer's algorithm to highlight the vector field topology. Motivated by the visual discontinuity in periodic orbits and separatrices in current techniques, before using the seed placement algorithm, they first extract periodic orbits and separatrices and make them the initial streamlines. Further, to avoid clutter near sources, sinks, and periodic orbits, they terminate a separatrix if it is within a user-defined distance from the non-saddle end.

Preceding the parallel hierarchical LTA algorithm presented in Section 4.1.1, Zhang et al. [ZD09] proposed to extract the flow topology and use the topological skeleton as the initial set of streamlines that segment the field. Next, additional streamlines are calculated by placing seed points at the center of topological areas in a recursive manner creating an approximately uniform distribution of streamlines. They extend the vector field domain in each di-

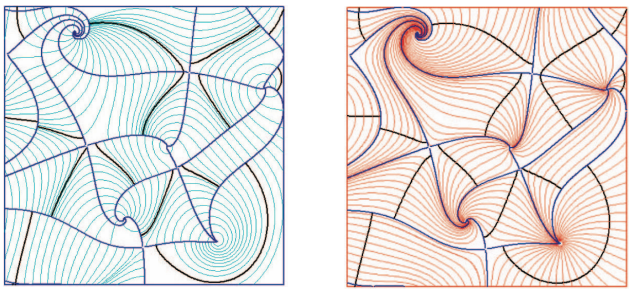

Figure 10: *Seed placement along the longest orthogonal curves (black) [WLZMI10]. Left: Orthogonal curves (light blue, black) and extracted flow topology (blue). Right: Orange streamlines are traced from seeds placed along longest orthogonal curves. Image courtesy of Wu et al.*

rection by adding a layer of mirrored boundary cells [ZS09]. Using the additional critical points from the extended vector field helps capture open separation and attachment lines.

Wu et al. [WLZMI10] similarly extracted the flow field topology and partitioned it into regions of uniform flow behavior. However, as opposed to adopting a recursive method, they search for the longest path that orthogonally crosses all streamlines within a region. Seeds are placed evenly along the longest path to produce approximately uniformly placed streamlines. They treat periodic orbits and saddle-connected loops as special cases. Using vector field reconstruction error as a quantitative measure for comparison, they demonstrate superior streamline placement than previous works [JL97, MAD05, LMG06, CML*07]. Their study shows that as a sparser set of streamlines is used, i.e., as the minimum separating distance increases, their algorithm results in lower reconstruction error in comparison to other algorithms. Figure 10 illustrates the use of the longest orthogonal path as a seeding curve.

Analysis: Flow topology-guided techniques have a primary objective of highlighting ROI in the flow field. With respect to redundancy, additional streamlines generated to fill empty spaces of the image or object space can introduce redundant streamlines given the use of only minimum separating distances. Computing the flow topology robustly for large scale complex vector fields is challenging given small changes in the flow field can lead to vastly different topological connections. However, for smaller-scale problems, computing the flow topology is tractable. The majority of these techniques are most appropriate for a planar or curved surface flow given they either do not extend to 3D or result in occluding streamlines in 3D. For 2D domains, these methods have been demonstrated to have low reconstruction errors, which is highly desirable for multiple applications.

4.2.2. Derived Field Techniques

The topological structure of the flow field often shows a strong correlation to fields that can be derived from the vector field. Thus, several research efforts have leveraged this property to propose the use of derived fields as an alternative approach to guide SPSS to capture salient flow features. The remainder of this section is divided into algorithms that use entropy, derived vector field characteristics, or user-defined scalar fields.

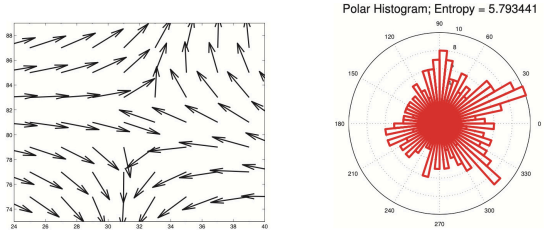

Figure 11: *Xu et al. measured entropy or information content in neighborhoods of the vector field [XLS10]. Left: an example vector field neighborhood. Right: the distribution of the vectors approximated using a polar histogram. Image courtesy of Xu et al.*

Algorithms Using Entropy (Information Theory)

We can measure the amount of information or uncertainty in a local region using Shannon's entropy $H(X)$, where X is a random variable that models the input vector field.

$$H(X) = -\sum_{x_i \in X} p(x_i) \log_2 p(x_i) \tag{3}$$

A second key concept is conditional entropy. If Y models the visualization output consisting of streamlines, the conditional entropy between two variables $H(X|Y)$ informs how much uncertainty in the input data X remains after a set of streamlines in Y are displayed. This measure allows more streamlines to be placed in regions whose information is not sufficiently captured. The entropy of a vector field can be computed for each grid point in the domain and provides insight into the variation among vectors for a given neighborhood. Further, the entropy of a streamline is the accumulated value of entropy at sample points along the streamline, and the entropy of a voxel is the average of entropy values at the grid points of a single voxel.

Furuya et al. [FI08] considered streamline selection in combination with scalar field isosurfaces. The algorithm begins by integrating a large number of streamlines. The entropy of segments of individual streamlines is measured and used as a basis for selection. The measure of streamline entropy accounts for occlusion caused by isosurfaces by penalizing a streamline if segments of the curve are occluded. Finally, streamlines are sorted and selected in order of highest entropy. To control density, streamlines are only rendered if they are some minimum threshold distance away from existing streamlines.

Xu et al. [XLS10] presented a framework that evaluates the effectiveness of a visualization by measuring how much information in the original data is being communicated. In this work, they empirically demonstrate that entropy in regions near critical points and separation lines is higher than that of other regions. Modeling a vector field as a distribution of directions, entropy is used to measure the information content in the vector field. Figure 11 illustrates the use of a polar histogram to capture the distribution of vector directions. The effectiveness of the streamline placement is measured by reconstructing a distribution of vectors derived from the selected streamlines. The approach begins by iteratively placing seed points in regions of high entropy. The authors use a diamond shape template to place 9 seed points in 2D and an octahedral shaped template consisting of 27 seed points in 3D. Further, to prevent large voids, seeds are placed in proportion to the computed conditional entropy.

Figure 12: *Seed placement stages of the entropy-guided algorithm by Xu et al. [XLS10]. Seed points are highlighted in red and the images show (from left to right) initial placement of seed templates, additional seeds used to reduce conditional entropy, a result of the algorithm, and visualization results of the Jobard and Lefer method [JL97] and Mebarki et al. algorithm [MAD05]. Image courtesy of Xu et al.*

The streamline addition process ends when the value of conditional entropy of the entire domain converges to a small value. A final pruning step removes redundant streamlines. Figure 12 shows the placement of seeds based on entropy, streamlines traced in two stages, and a comparison of the selected streamlines for the proposed approach to previous works [JL97, MAD05]. The algorithm provided quantitative control of the selection of streamlines.

Lee et al. [LMSC11] extended the framework presented by Xu et al. [XLS10], to support a view-dependent streamline selection aimed at minimizing occlusion and revealing important flow features. Using the derived entropy field, a maximal entropy projection (MEP) frame buffer is computed for a given image space. The MEP buffer stores maximal entropy values, as well as the corresponding depth values for the given viewpoint. To identify the optimal viewpoints, i.e., views that convey maximum entropy information, MEPs of 780 viewports are evaluated. A streamline is assigned a higher priority if it reveals the flow near salient features and a lower priority if it occludes an important region of the flow. Streamlines are segmented and each fragment in the image plane is evaluated using information stored in the MEP buffer, i.e., depth and entropy, to compute a scalar score ω. The streamlines are prioritized based on their value of ω which can be positive or negative. To maintain a streamline density proportional to the flow complexity, the screen space is divided into tiles and an expected streamline density equal to the average normalized entropy of the region in the MEP buffer is computed. For a given streamline, if the addition of the streamline affects more tiles with a density lower than the expected density, the streamline is added. This approach favors streamlines that reveal salient flow features without occluding other more important features. They demonstrated improved feature capturing compared to Marchesin et al. [MCHM10] for a view-dependent selection of streamlines. Further, although they significantly benefit from using a GPU, the serial selection is a bottleneck.

Ma et al. [MWW*14] present FlowTour, a framework that selects best viewpoints to explore a flow field visualized using streamlines. A skeleton-based seeding algorithm is employed to generate a set of streamlines that capture critical regions of the field. Variation of both direction and magnitude of vectors are considered to compute the entropy of voxels. Critical regions are identified as local neighborhoods in which voxel entropy values exceed a threshold. Sufficiently large regions of connected voxels with high entropy are used as input to a volume thinning algorithm that extracts the skeleton points. Skeleton points are connected by applying a minimum spanning tree algorithm to produce a tree-structured skeleton line. The density of streamlines is controlled by evenly-spaced seed placement along the skeleton. Candidate viewpoints are generated on the basis of the critical regions identified in the field. Finally, the best viewpoints for each region are selected and connected into a view path using a B-spline curve.

Algorithms Using Derived Vector Field Characteristics

The problem of image space cluttering is exacerbated in 3D unsteady state flow, given pathlines can intersect in space. To study unsteady state 3D flow, Wiebel et al. [WS05] introduced the concept of an eyelet. Calculating a set of pathlines that pass through the same single point (eyelet) in space at different times yields an insightful static visualization of the unsteady flow field. The collection of pathlines can be used to construct a surface to visualize the flow. If pathlines diverge more than a user-defined threshold, a new seed is added at the eyelet at a time step in between the time steps of its neighboring particles. While this approach is conceptually similar to Hultquist [Hul92], instead of adding a new point at the location where divergence is detected, it is added at the eyelet to guarantee the particle would indeed pass through the eyelet. The placement of the eyelet plays a critical role in the usefulness of the algorithm to study the flow field. They identify regions of high activity by introducing measures to capture the change of a vector field over time. A dot product variation is computed by accumulating the positive dot products of vectors in consecutive time steps. A second measure, the vector variation is the norm of the computed difference of the two vectors considered. Isosurfaces drawn using these variation fields help identify regions of high and low activity for further investigation. Additionally, edges and corners, or regions behind flow passed objects, singularities, and vortex cores serve as good locations to place an eyelet.

Wang et al. [WZN11] visualized explosion fields by first generating an isosurface of the magnitude of velocity. The isosurface region is then divided into a series of subregions that are almost equal in area. The center of each subregion is used as a seed point with the integrated streamlines always starting from the center of the explosion and extending outward in a direction perpendicular to the isosurface.

Luo et al. [LSW12] proposed a technique that combined the use of derived scalar fields and topological methods, such as contour trees and persistent homology. Global importance is measured in terms of persistence of topological features in the vector field and streamline density in the generated visualization is used to reflect the same. Hodge decomposition [AMR12] is a technique used to convert a 2D vector field into two scalar fields gradient potential and curl potential. Maxima and minima of the gradient field correspond to sources and sinks respectively and guide the placement of a set of *gradient* seeds. The number of seeds placed is propor-

tional to the persistence of the maxima and minima, measured by determining the amount of perturbation required to smooth out the mountain peak or valley. A contour tree encodes the evolution of level sets of the curl field and is used to generate a set of *curl* seeds. Each branch of the contour tree corresponds to a topological component of the domain. Each branch is assigned a number of seeds proportional to its range function to collectively produce a set of curl seeds. Every seed location is evaluated to determine gradient vector or curl vector magnitude dominance at that position. Only gradient seeds in positions of gradient dominance are used. Similar, only curl seeds in positions of curl dominance are used. Luo et al. demonstrated superior placement quality in terms of reconstruction error compared to Li et al. [LHS08] and Xu et al. [XLS10].

Yu et al. [YWSC12] used the curvature and torsion of the vector field to generate a saliency map to guide seed placement. The saliency map is computed as the difference between Gaussian-weighted averages of curvature and torsion fields calculated at multiple scales, i.e., varying the standard deviation of the Gaussian filter. The saliency map is computed for five threshold distances and then all five saliency maps are combined with a nonlinear normalization. While the computation of the saliency map is relatively expensive on CPUs, it can be computed within a few seconds on a GPU. Given a final saliency map, the seed placement algorithm selects locations in order of decreasing saliency. Long streamlines are favored, with streamlines integrated until they reach a critical point or leave the domain. To reduce redundancy, streamlines that occupy the same voxel as an existing streamline are discarded. The use of the saliency map as opposed to directly seeding based on the curvature or torsion fields, allows streamlines to be placed closer to critical points. Further, the generated set of streamlines is hierarchically clustered to enable exploration at different levels of detail and manage clutter (details in Section 4.3.1).

Zhang et al. [ZCL*16] investigated the usage of a scalar field Φ derived from the input vector field by integrating the rotation of the integral curves. Seeds are placed where $|\nabla\Phi|$, the magnitude of the rate of variation of the derived field Φ, is greater than a user-defined threshold. Randomly starting from a placed seed point, an integral curve is computed, followed by the filtering of nearby seeds. The process is repeated with the remaining seeds.

To visualize a vortex rope that builds up in the draft tube of a water turbine, Bauer et al. [BPSS02] proposed a particle seeding scheme to visualize unsteady flow. They use Sobol quasirandom sequences [SH95a] to obtain a uniform distribution while avoiding clustering and artifacts like regular patterns. Given the vortex rope is a rotating helical structure, the helicity in the field is evaluated to identify ROI. New particles are introduced to regions with a scalar value of helicity greater than a predefined threshold by offsetting the original point set of quasirandom sequences. They use a layer of invisible buffer cells that enable particles to fade in and out smoothly from the ROI. Guthe et al. [GGS02] presented another seed placement approach aimed at distributing more particles in ROI. The seed placement is based on an adaptive sampling of the field with the goal of achieving a higher sampling resolution in more interesting regions and a lower sampling resolution in less interesting regions. The local gradient, divergence, and curvature of the vector field are used to influence the particle distribution. Additionally, local shear and rotation of the vector field or distance to the closest critical point can be used. An octree data structure is employed to maintain the distribution of particles in the domain. The distribution octree is updated as particles travel along streamlines with particle age being a deciding factor in regard to particle removal in overcrowded regions.

Particle-based visualization systems have seen efforts to improve the interactivity of flow visualization [FG98, KKKW05, BKKW08, VPBB*11]. Engelke et al. [ELPH18] proposed a particle system that results in an adaptive particle density by using autonomous particles. Particles operate in parallel without neighborhood information or inter-particle communication by following a set of rules. The rules dictate particle birth, death, and split events that influence the density of particles in different regions of the flow. The study uses split criteria such as λ_2 [JH95], the curvature of the particle trajectory, and distance to an object in the field. The parallel nature of the system allows interactive visualization while maintaining a smart sampling of the flow. Both context and feature particles are used, with context particles being randomly introduced into the domain to prevent underrepresented regions. Feature particles are children of context particles and are introduced when a split event occurs. Split events are determined by a combination of properties such as energy, generation of the parent particle, and the local importance measures in the flow.

Analysis: Several studies demonstrated the use of derived fields such as entropy, curvature, and torsion to highlight salient features of a flow field using streamlines. Additionally, derived fields provided a means to visualize regions of maximum interest irrespective of the depth from a given viewing angle. To reduce redundancy the use of a pruning step to remove redundant streamlines was demonstrated, however, minimal redundancy is not the focus of these algorithms. Deriving fields from the vector field can often be performed in parallel and computed relatively fast without considerable scalability issues. Thus, these techniques have the benefit of fast computation and the ability to highlight ROI. Considering these benefits, derived field techniques have the potential to be applied to particle-based flow visualization systems or within in situ flow visualization contexts.

4.2.3. User-Defined Scalar Field Techniques

The algorithms in this section are designed to use either a specific user-defined scalar field or are adaptable to utilize any given scalar field to generate a representative set of streamlines.

Zockler et al. [ZSH96] proposed to use a statistical method to facilitate the placement of streamlines with a density proportional to some scalar quantity. Considering a uniform grid over the domain, for each cell, a local degree of interest is computed. Cells are selected on the basis of the parameterization of a probability distribution using the local degree of interest. Seeds are then placed in those cells and streamlines are forward and backward integrated for a fixed length. For scalars ranging over multiple orders of magnitude, streamline distribution can be unsatisfactory. To address this problem, a histogram equalization approach is used to obtain a homogenous distribution. Weinkauf et al. [WT02, WHN*03] applied this technique using fields of curvature and torsion.

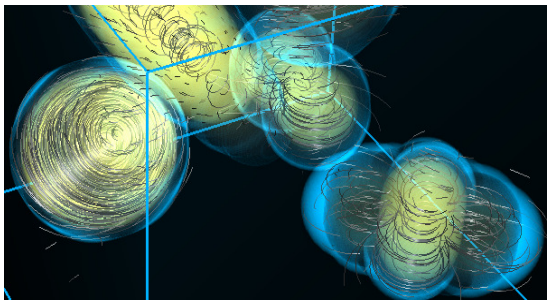

Figure 13: *Streamline visualization of vortical behavior [BKH*15]. Image recreated.*

Schlemmer et al. [SHH*07] presented a heterogeneous distribution of streamlines based on a density map derived using scalar fields (temperature), derived vector field information (magnitude of velocity, vorticity), or a user-defined density function. Here, streamline density is the number of occupied cells over the total number of cells in a domain. To calculate *priority* streamlines, they first define a density map used to guide seed placement, with the map updated after every streamline calculation. The first seed is placed at the location of the maximum value of the initial density map. The next location is chosen as the furthest of the next five maximum values. Given the density map is monotonically decreasing over time as streamlines are added, the algorithm will eventually terminate.

Shen et al. [SVW16] proposed the use of fractal dimensions [KW10] for streamline selection. Measured using the box-counting ratio [KW10], fractal dimensions can provide insight into the complexity of a streamline by considering its space-filling properties [CLSW14]. The box counting ratio is measured by counting the number of cells a streamline intersects in a small grid. For each voxel in the domain, a scalar value is calculated using the local box counting ratio of streamlines that intersect the voxel. The scalar grid is used to filter streamlines by fractal dimension and to identify regions containing vortices and turbulence. Although the space-filling properties of streamlines are evaluated and desired regions are highlighted, it does not guarantee to capture ROI in the flow. The algorithm demonstrates the ability to capture a feature focused set of streamlines and remove redundant curves.

Given a pattern template as input, Bujack et al. [BKH*15,BH17] derived a scalar field that shows how similar each location in the vector field is to the template using rotation-invariant pattern detection. Next, they seed streamlines with a probability that is proportional to the calculated scalar field. This method is able to explicitly visualize ROI as defined by the user and minimize their occlusion. Figure 13 shows a 3D example field with streamlines only seeded in areas of vortical behavior.

Analysis: Algorithms using alternative scalar fields defined by the user are capable of highlighting ROI to the user. However, these are less traditional approaches to visualizing flow features and require users to define parameters. Given the diversity in the underlying approach, these methods demonstrated varying degrees of control with regard to minimizing redundancy. Pruning of streamlines can be easily integrated into a technique by controlling the number of streamlines passing through any cell. Lastly, these techniques can be relatively fast given the easy generation and use of a guiding scalar field in conjunction with straightforward algorithms.

4.3. Similarity-Based Techniques

SPSS techniques based on density distribution or feature extraction have certain drawbacks, i.e., redundancy, require feature extraction or derivations of guiding fields. Similarity-based approaches have been gaining popularity in the past decade due to their ability to overcome these drawbacks and by serving multiple flow exploration tasks. These techniques are based on the concept of first identifying similar streamlines and then selecting representatives from groups of similar streamlines. Additionally, these approaches support hierarchical grouping of streamlines for a level-of-detail approach and the identification of streamlines similar 2to a query streamline, i.e., isolating streamlines that match a given description. Similarity between streamlines is measured using spatial proximity (4.3.1), shape (4.3.2), or by employing machine learning (4.3.3) to cluster streamlines based on several feature attributes.

Streamline clustering and selection research has extended beyond the tasks mentioned above to include feature-specific selection, application to non-dynamic vector fields or medical applications, and methods to improve the costs of clustering. For example, in an effort to highlight structures of interest in the flow, Salzbrunn et al. [SS06, SGSM08] define streamline and pathline predicates to cluster similar integral curves that satisfy some criteria. Clustering of curves is commonly used to visualize diffusion tensor imaging (DTI) data [BKP*04, MVVW05, OW05, TWHW07, MGWW08]. With respect to medical visualization, Oeltze et al. [OLK*14] evaluated three clustering techniques — k-means, agglomerative hierarchical clustering (AHC), and spectral clustering to reduce clutter when visualizing streamlines traced from simulated blood flow. Recently, Shi et al. [SLC19] conducted an in-depth comparative study of several curve clustering and simplifications algorithms used for flow visualization to provide users with a systematic guideline to choose a specific approach. Lastly, given the high cost of similarity metrics that involve pairwise streamline comparison, Shi et al. [SC17] proposed metrics that run in linear complexity.

4.3.1. Spatial Distance Techniques

Streamlines in the proximity of one another are likely to have sections that display similar curves. One way to identify such streamlines is through spatial distance. The remainder of this section is divided into algorithms that use proximity as a measure of similarity or methods that use the mean of closest point distances as a similarity metric.

Algorithms Using Dissimilarity Metrics

Motivated by the shortcomings of density and feature guided methods, Chen et al. [CCK07] presented the first similarity guided streamline placement algorithm for 2D and 3D steady flows. The algorithm naturally accentuates regions of geometric interest while minimizing streamlines in areas of parallel flow. As a measure of similarity between two streamlines, a similarity distance metric that has two influencing factors is defined. The first factor is a translational distance measured as the Euclidean minimum distance between points on two streamlines. The second factor is a measure

of shape and orientation similarity and is measured over a spatial window. A spatial window is formed by identifying a predefined number of equally spaced sample points along the curve. The translational distance is the average deviation of sample point pair distances from the the center point pair distance. Starting from a dense set of candidate seed points, a streamline is traced until its similarity distance for a window falls below a pre-specified similarity tolerance. If the streamline length is greater than a minimum length threshold, the streamline is added to the set of selected streamlines. The use of shape and orientation for similarity results in 30% better placement compared to only using the translational distance. However, the method is sensitive to the order in which candidate seed points were tested.

Li et al. [LHS08] proposed an iterative algorithm to select a small set of streamlines in 2D steady state flow fields. The algorithm exploits the spatial coherence in a flow field to achieve a minimal selection of representative streamlines. The density of selected streamlines, each of which is integrated for as long as possible, varies to reflect the different degrees of coherence in the field. Their algorithm derives local and global metrics by employing 2D distance fields that measure the distances from each grid point to nearby streamlines. The local metric measures the direction difference between the vectors of the original field and an approximate field computed from streamlines in the vicinity. The global metric measures streamline dissimilarity by accumulating the local dissimilarity at every integrated point along a streamline trajectory. To place seed points, the algorithm begins by placing a random or central seed point. Next, the streamline is integrated and local dissimilarity is evaluated at each grid point. A streamline is accepted if the local dissimilarity value of the original seed point is greater than a threshold and if the global dissimilarity value of the streamline is greater than a second threshold. The next candidate seed is picked by sorting the grid points into a sorted queue in descending order of the local dissimilarity value. The process ends when no remaining candidate seeds satisfy the dissimilarity threshold requirements. To optimize the algorithm, the number of candidate seeds is reduced by eliminating grid points on boundaries and by marking cells visited by rejected streamlines, i.e., streamlines with global dissimilarity values below the threshold.

Algorithms Using Mean of Closest Point Distances

The mean of closest point distances (MCPD) [MVVW05] is the mean of Euclidean distances between pairs of points formed by mapping each point of one streamline to the closest point of the other. MCPD has proven useful for multiple SPSS algorithms.

Building on the feature highlighting streamline generation technique presented in Section 4.2.2, Yu et al. [YWSC12] enable exploration at varying levels of detail via a hierarchical streamline bundling visualization. To calculate the bundles of streamlines, i.e., clusters, MCPD is used as a similarity measure. Beginning with each streamline in a distinct cluster, they successively merge the two most similar streamlines in a bottom-up fashion until a stopping criterion is reached. Clusters are merged using the *single-link* method where the distance between two clusters is the minimum of the distances between all pairs of member streamlines. To represent each cluster of streamlines, as opposed to streamlines close to the cluster centroid, the union of streamlines along the cluster boundary is used. In addition to capturing sources and sinks, only boundary streamlines of a cluster best reveal the saddle together with other boundary streamlines of neighboring clusters. Figure 14 illustrates a result of their clustering visualization.

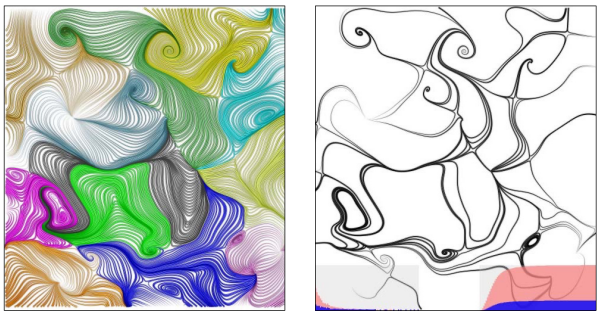

Figure 14: *Results of the hierarchical bundling visualization algorithm [YWSC12]. Left: hierarchical clustering of the flow field using 12 (top) and 70 (bottom) clusters. Right: A representative set of streamlines. Boundary streamlines of clusters are used and they highlight saddles in the flow. Image courtesy of Yu et al.*

While Yu et al. [YWSC12] used MCPD as a measure to merge streamlines into a cluster, Tao et al. [TMWS13] use the measure to identify redundancy and limit the number of representative streamlines selected. Measured using a *streamline information* metric, the selection is performed by considering the contribution of a streamline to a large set of sample viewpoints. Starting with a random pool of streamlines, a matrix containing the probabilities of seeing each streamline from each viewpoint considered is created. The probability of seeing a streamline is high if it contains a high amount of information in 3D and the 2D projection for a given view preserves the information well. Additionally, they score the shape characteristics of a streamline projection by evaluating each segment of a streamline subsampling and score segments higher if they form a 45 or 135-degree angle to the viewing direction. *Streamline information* represents the degree of dependence between a streamline and the set of viewpoints. A low value indicates a streamline contributes in a balanced manner to a large number of viewpoints, while a high value would indicate a streamline visible in a small set of viewpoints. Streamlines are sorted into a priority queue in decreasing order of accumulated streamline information. However, these streamlines have significant redundancy and are likely to cause clutter in 3D. The pairwise similarity between streamlines is measured using MCPD to avoid repetitive streamlines. Further, a *viewpoint information* measure is defined to similarly guide viewpoint selection for the chosen streamlines. In comparison to other works, the vector field reconstruction error for this algorithm is lower than both Xu et al. [XLS10] and Marchesin et al. [MCHM10]. Han et al. [HTZ*19] employ this technique in situ to save a compressed representative set of streamlines for post hoc flow analysis.

Opacity adaption is a technique used to manage occlusion and cluttering of streamlines in a 3D field. However, it can result in the loss of spatial perception when streamlines are faded out to reveal ROI. To retain the perception of spatial relationships between streamlines, Kanzler et al. [KFW16] select a set of streamlines such that the screen-space density of the streamlines is locally adapted

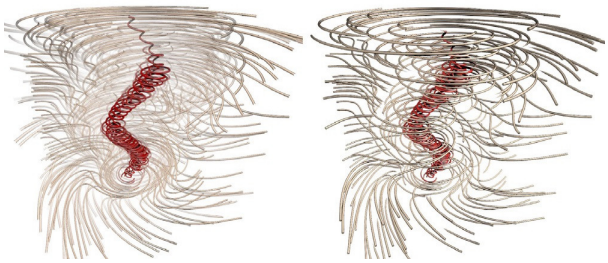

Figure 15: *A result from using the balanced line hierarchy approach to preserve spatial perception [KFW16]. Left: opacity adaption [GRT13]. Right: balanced line hierarchy and visibility values based on importance. Image courtesy of Kanzler et al.*

to the importance of the streamlines. Figure 15 shows an example comparing the use of opacity optimization and adaption of screen-space density. The algorithm requires computing a fully balanced line hierarchy to facilitate the uniform removal of streamlines in the domain and obtain the desired density at run time. MCPD is computed for all pairs of streamlines and is used to define a fully connected distance graph. A minimum cost perfect matching algorithm is recursively used to identify pairs by minimizing the sum of all included edge weights, i.e., the similarity measure. Single linkage is used to merge clusters since it results in a spatially coherent merging of clusters. Streamlines are selected by using visibility values based on (1) an importance measure, such as curvature, measured along the streamline, and (2) the occlusion caused by the streamline to other potentially more important lines. Visibility thresholds are assigned to lines in the hierarchy based on the level at which the line is the representative for its cluster. Further, the visibility values are then used to locally control the line density. Although this method improves the spatial perception of the visualization, it incurs a long preprocessing time to build a balanced line hierarchy.

Analysis: Algorithms using spatial distance to measure the similarity between streamlines were capable of selecting distinct streamlines that accurately captured flow features and provided a parameter to control redundancy. Calculating the similarity between large numbers of streamlines can be computationally expensive and require large preprocessing times depending on the approach adopted. The shortcoming of using a spatial distance for similarity measure is that the metric is limited to the similarity between streamlines in proximity, i.e., similar streamlines present in different regions of the flow would not appear similar. This limits the use of spatial distance to the application of generating a representative set of streamlines.

4.3.2. Shape Techniques

Feature attributes have been extensively used to evaluate the similarity between streamlines. A common use case is to identify all streamlines similar to a given streamline. Distance-based similarity measures primarily account for proximity and are sensitive to rotation, translation, and scaling. Feature attributes address these shortcomings by evaluating similarity in a proximity-, size-, and orientation-insensitive manner. The remainder of this section is divided into algorithms that use sample points or segments of a streamline to measure shape similarity between streamlines.

Algorithms Using Point Sampled Features

Wei et al. [WWYM10] proposed a technique to select streamlines similar to a user-sketched streamline. The user sketch is a 3D curve whose 2D projection is used as the input to the algorithm. The algorithm approximates the sketched curve and the streamlines using an arc length parameterized cubic B-spline and samples the curvature at equal arc length intervals along the curve. Using a feature vector constructed by concatenating the curvature and torsion at sampled points, they employ a *string matching* approach to find similar streamlines. The difference between two feature vectors is measured using the *edit distance* [WF74]. The most similar 3D streamline is identified and used it as a reference for clustering. All streamlines similar to the reference streamline are selected as the result of the streamline query. Additionally, Wei et al. proposed to choose cluster representatives from an AHC scheme based on view-dependent quality. Viewpoint quality of a streamline is computed by accumulating the winding angle of the 2D projection of the streamline, with larger values representing more information.

Zheng et al. [ZWL15] presented a streamline selection algorithm for 2D flow fields that uses streamline feature classification, similarity, and entropy. Streamlines are first classified based on the feature they highlight, i.e., a vortex, source, sink, or saddle, and are also prioritized in that order when they capture more than one feature. In addition to feature type, each streamline has a feature position, i.e., the vortex center for a vortex streamline, the critical point for a source-sink streamline, or the point of highest entropy for a saddle streamline. Next, streamlines are iteratively clustered on the basis of a combination of feature type and proximity of the feature position. A similarity metric that uses a combination of both geometric shape properties and proximity is defined. Curvature and accumulated angle are used as geometric shape properties. Sample points between two streamlines are mapped using Dynamic Time Warping (DTW) [BC94], which is a dynamic programming algorithm to find an optimal mapping between two sequences. For a proximity evaluation, MCPD is employed. The algorithm selects a set of streamlines by identifying a streamline from each feature subset with the highest entropy accumulated along the points of the streamline. Streamlines within each feature subset that are least similar to the previously selected streamlines are picked next. The last step involves, selecting more streamlines by considering similarity to previously chosen streamlines and streamline entropy until a desired number of streamlines is selected. The algorithm limits the number of redundant streamlines and is capable of generating a placement qualitatively equivalent to the work by Yu et al. [YWSC12] for a 2D flow.

Algorithms Using Segmentation

To tackle the high computational expense of distance-based similarity measures that involve performing large numbers of Euclidean distance tests, McLoughlin et al. [MJL*13] propose to measure streamline similarity by first computing an integral curve-specific signature. The signature is computed by segmenting an integral curve and using a set of curve-based attributes, namely, curvature, torsion, and tortuosity, to describe the integral curve per unit length of the curve. The χ^2 test [Pea00] is used to measure the similarity between streamlines and is performed for all streamline pairs generating a similarity matrix that enables fast lookup

for the clustering process. Additionally, given the streamline signature is proximity independent, a Euclidean distance measure can be used to supplement the χ^2 similarity measure. To address segment alignment issues when comparing a pair of streamlines, seed placement is limited to rakes orthogonal to the local flow and a hierarchical signature for each streamline is considered. The algorithm was demonstrated to be orders-of-magnitude faster than the approach by Chen et al. [CCK07].

Chen et al. [CYY*11] use an entropy-guided seed placement strategy to generate an initial set of streamlines. Streamlines are clustered using a two-stage k-means algorithm. The first stage only considers the start, middle, and end points of a streamline for clustering. Each cluster after the first stage is further subdivided into clusters by considering the linear and angular entropy of streamline segments. The two-stage k-means algorithm is chosen over single-linkage clustering with MCPD due to the quadratic computational complexity of the latter.

Lu et al. [LCL*13] proposed a similarity measure based on the statistical distribution of measurements along a streamline. As a result of being based on distributions, the similarity measure is less sensitive to length, spatial location, and orientation. First, streamlines are recursively segmented until a minimum length threshold is reached or a segment cannot be split into two segments which are dissimilar enough. Next, a 1D histogram is constructed to represent every segment, and a 2D histogram to represent the entire curve. The use of the 2D histogram is to capture the order of the segments, and thus avoid dissimilar streamlines with similar feature distributions appearing similar. Given streamlines may be represented by a varying number of segments, a mapping between two sets of segments is performed by using DTW. The difference between two histograms is measured using earth mover's distance (EMD). Curvature, torsion, and curl are used as measures along a streamline to demonstrate the distribution-based approach. Their proposed AHC scheme, in addition to using a distance measure (for example, *single-linkage*), uses a balance parameter that accounts for the number of streamlines in a cluster and can force smaller clusters to merge early in the process to produce a more balanced tree. The algorithm was demonstrated to be much faster than using distance measures to identify similarity.

Li et al. [LWS14] proposed to use a similar feature vector description approach to measure streamline similarity. To address the issue of dissimilar streamlines having very similar feature distributions, they use a feature descriptor that encodes the spatial relations among the features. The encoding mitigates the need to segment the streamline and find mappings between two streamlines during similarity evaluation. Besides using local streamline metrics like curvature and torsion, Li et al. use global geometric properties of tortuosity and velocity direction entropy to describe a streamline. A weighted Manhattan distance between constructed feature vectors of two streamlines measures the similarity between them. Following a pairwise similarity evaluation between all streamlines, *affinity propagation* [FD07] is used to cluster streamlines and form a hierarchy. The affinity propagation algorithm accepts the measured similarity values as input and simultaneously considers all the data points as possible cluster centers. It uses similarity values as preference values for each data point. The algorithm then exchanges real-valued messages between data points until it converges to produce a set of cluster centers of high quality.

Reference	Similarity Measure	Clustering
[CYY*11]	Spatial, Shape Properties	K-means
[MJL*13]	χ^2	AHC
[LCL*13]	EMD	AHC
[LWS14]	Manhattan Distance	Affinity Propagation
[TWS14]	Procrustes Distance	Affinity Propagation

Table 1: *Similarity measures and the corresponding clustering methods used by similarity-based techniques that use segmentation.*

FlowString is a framework for partial streamline matching proposed by Tao et al. [TWS14] that models streamlines as strings. Streamlines are first resampled on the basis of winding angle, using a threshold small enough to capture relatively simple patterns of the streamline segment between neighboring sample points. All sample points are then evaluated pairwise for a similarity measure, the *Procrustes distance* (a metric to quantify similarity for 3D shapes and is extensively used in biological morphometrics), followed by applying *affinity propagation* for clustering using a GPU. The resultant clusters after two levels of affinity propagation serve as the local shapes for the data set. The local shapes are used as characters, which together form an alphabet, using which words can be formed by concatenating characters together. Their work differentiates between approximate and exact searches; they achieve the former with dynamic programming and the latter with a suffix tree. This work was the first to pursue labeling and classification of streamline segments.

Analysis: Use of features along a streamline allowed for comparison between streamlines in a proximity-, scale-, and orientation-invariant manner. Table 1 shows similarity measures and corresponding clustering techniques used. Streamlines were either identified by the feature attributes of points along the curve or curve segments. Thus, with respect to the ability to identify ROI, these algorithms are capable of responding to streamline similarity queries in addition to calculating a representative set. In general, similarity-based methods provide the user with control of redundancy by allowing the number of clusters or similarity threshold to be varied. However, computing similarity between streamlines requires a one-to-one comparison and significant processing times. To accelerate the process, there are two major techniques: (1) using accelerators for clustering, and (2) use of segmentation to avoid large numbers of Euclidean distance checks. However, these similarity measures are being performed on a relatively small number of curves and extending these techniques to scale by using more computationally efficient measures is a current research area.

4.3.3. Machine Learning Techniques

The application of machine learning techniques to scientific visualization problems has been a recent development in the field. With respect to flow visualization and specifically the use of streamlines, there has been recent activity.

Algorithms Using SVM For Segmentation

Streamline segmentation has growing importance given its application in identifying the similarity between streamlines. However, current measures of segmentation and similarity measurement do

not account for human perception or what a human considers important. Li et al. [LWS15] adopted a user-guided approach that used a binary support vector machine (SVM) to perform streamline segmentation. The approach begins by first generating a pool of random streamlines, followed by the use of affinity propagation for clustering based on a similarity metric that uses curvature and torsion 1D histograms. 1D histograms, formed by concatenating previously computed histograms of curvature (20 bins) and torsion (40 bins), describe the shape characteristics of a streamline. Next, users choose segmentation points along the set of representative streamlines from each cluster. For every segmentation point, the algorithm computes a feature vector comprising velocity direction ratio, tortuosity ratio, the curvature and torsion histograms, and the volume ratio of minimum-bounding ellipsoids using varying neighborhood sizes. User-selected segmentation points are positive training samples, while all non-segmentation points are negative training samples used to train an SVM classifier. The streamline segmentation process is carried out for the remaining streamlines using the classifier. If a group of nearby points is selected as segmentation candidates, a post-processing step chooses the point with the smallest ratio of minimum-bounding ellipsoids as the final segmentation point. The algorithm presented by Li et al. is the first to use supervised machine learning for streamline segmentation. They demonstrated superior segmentation and feature capturing in comparison to previous similarity-based methods that used segmentation.

Algorithms Using DNN For Feature Description

Han et al. [HTW18] used a DNN, named FlowNet, to identify a representative set of streamlines for a given flow field. An autoencoder, which features both convolutional and fully-connected layers, enables the network to learn a complex data representation by using both local and non-linear combinations of neurons. FlowNet accepts voxelized and downsampled representations of streamlines as input to learn features by non-linearly mapping each representation to a feature descriptor of 1024 dimensions. A binary cross-entropy loss function is employed to train FlowNet. To explore the feature descriptors generated, t-SNE [MH08] is applied for dimensionality reduction, followed by interactive parameter tuning of the clustering method DBSCAN [EKS*96] to find a suitable number of clusters or set the minimum number of samples in a cluster. A representative streamline is then identified as one who minimizes the sum of Euclidean distance to all other points in the cluster. Han et al. don't explicitly use any streamline attributes, but instead, are the first to employ a DNN to calculate flow features before clustering and selection. They demonstrated streamline selection which results in lower vector field reconstruction error (Table 2) compared to Tao et al. [TMWS13] and Xu et al. [XLS10]. However, training the network can require days to complete.

Analysis: The use of machine learning for scientific visualization has increased recently. Machine learning has been leveraged to accurately capture ROI in a volume flow and to perform streamline segmentation based on a user specification of what is considered interesting. The method by Han et al. [HTW18] selects only representatives of clusters and thus minimizes redundancy. Currently, the two major drawbacks of these methods are (1) the need to subsample the data set in order for it to be feasible to process and (2) long run times. Further, the features learned are data set-specific and require each data set to be processed. Future efforts can aim to build a large database to train neural networks to identify flow features. Thus, there is potential for further research concerning the application of machine learning techniques to unsteady flow visualization and overall computational improvements.

Dataset	[HTW18]	[TMWS13]	[XLS10]
crayfish	**0.102**	0.116	0.144
solar plume	0.283	**0.280**	0.303
five critical pts	**0.023**	0.026	0.031
tornado	**0.080**	0.105	0.101
two swirls	**0.065**	0.070	0.071

Table 2: *Vector field reconstruction error measures from the study by Han et al. [HTW18]. Error is measured as the average angle difference between the original vector field and the vector field reconstructed using the streamlines.*

5. Manual Techniques

Manual placement of initial seed positions is a common first step when using streamlines to study a flow field. Interactive flow visualization techniques were introduced two decades ago and have since evolved to give the user varying degrees of control of the generated visualization. While several interactive flow visualization techniques exist, in this section, we limit our study to manual techniques involving tools for placement of seeds, density control, and the use of domain information to do the same.

5.1. Use of Interactive Tools

The virtual windtunnel project [BJS*98] was an immersive virtual reality-based system used for the investigation of airflow around a Space Shuttle. In this work, Bryson et al. describe the use of a hand position-sensitive glove controller for injecting particles into a 3D unsteady flow. To study ROI in the flow, such as boundary layers and turbulent regions, the locations of seed points are interactively selected and the corresponding pathlines are integrated. The resultant graphics objects can be visualized and manipulated. In addition to rapid seed placement, the environment supports the repositioning, grouping of seeds as a rake, and deletion of existing seed points using hand gestures. Hardware limitations at the time (1998) meant spatial subsampling of the vector field was required for interactivity or the use of a supercomputer with dedicated graphics resources.

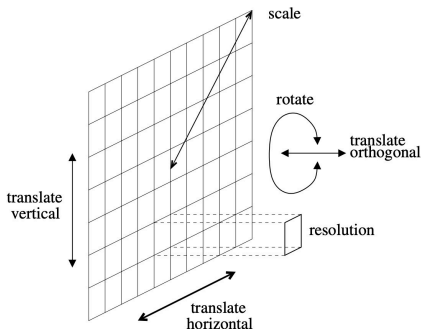

Figure 16: *Seeding plane to manually place seed points in a 3D flow field [LWSH04]. The plane can be oriented and scaled as needed. Image courtesy of Laramee et al.*

Schulz et al. [SRBE99] aimed to improve the interactivity of a virtual reality-based exploration system. The target application, a car body development aerodynamics simulation, required particle tracing to account for collisions with the car body. The initial positions of seeds are specified using a freely movable probe similar to Bryson et al. [BJS*98] and aligned on a rake or inside a cube. Additionally, they proposed application-specific data structures and interpolation techniques for fast particle tracing.

Laramee et al. [LWSH04] proposed the use of a manual seeding tool which allowed six degrees of freedom. The seed placement tool, shown in Figure 16, is a two-dimensional seeding plane grid. Initial seed locations are set at the grid points before streamlines are calculated. Besides offering grid resolution control, the seeding plane can be translated, rotated, and scaled enabling convenient seed placement options. Laramee [Lar02] proposed another system, named Streamrunner, which attempted to address problems of occlusion, and the lack of directional and depth cues when interactively using streamlines in 3D flow. Streamrunner gave the user control over seed placement and the evolution of streamlines from the time they are seeds until they reach full length. This provided users with a sense of direction of flow and depth when observing the growth of the streamlines.

5.2. Use of Domain Information for Direct Seed Placement

To generate informative visualizations, the manual placement of seed points or rakes is most suitable when aspects of the flow field behavior are known and scientists can intelligently place seed points. Alternatively, seed points may be manually placed near local maxima of an interesting scalar derived from the vector field or near critical points of the vector field topology [SH95b]. Several flow visualization works adopt this approach.

For certain applications, knowledge of moving objects in the domain or specific component design assists in deciding seed point locations. Engineers are often required to evaluate the pattern of flow, such as a swirl or tumble flow, in the combustion chamber of an automobile in order to achieve efficient and stable combustion. Laramee et al. [LWSH04] strategically placed a seeding plane near the intake ports of a combustion chamber from where fluid enters to evaluate swirl flow. Multiple seeding planes are manually placed and the length of the generated streamlines in the combustion chamber to capture tumble motion is limited. In another study involving the complex geometry of an automotive engine cooling jacket, Laramee et al. [LGD*05] generated seed points at the inlet of the cooling jacket. To maintain seed density, a scheme similar to Bauer et al. [BPSS02] is adopted and particles travel along integral curves until they hit a boundary or leave through the outlet. Interactively exploring the cooling jacket proved tedious given the rapid visual clutter created by complicated twisted paths and the difficulty in identifying recirculation zones. To visualize flow past a marine turbine, Peng et al. [PGN*13] used derived swirl flow information and multiple-coordinated views to assist domain experts manually place seeds in regions of reverse flow.

For a biology-inspired CFD simulation, Koehler et al. [KWDG11] presented a novel seed placement method for the visual flow analysis of insect flight. The domain contained multiple dynamically deforming flapping dragonfly wings. Traditional methods of using static seed points suffer in situations where there are immersed boundaries in the flow field. The technique is based on the premise that interesting flow phenomena generally occur near and move with the wings. Seeds are bound in the direction of the vertex normal of user-selected points near the surface of the immersed objects (for this application the wings). The user is given control of the seed density and how far in the normal direction seeds are placed. Seed curves are obtained by connecting points at neighboring time steps that are the same distance in the normal direction of the same point on the wing mesh. Seed curves are then color-mapped to a scalar of interest such as velocity, vorticity or λ_2. The user can then choose seed curves that are informative and be used further to generate various integration-based flow lines. Figure 17 illustrates the seed placement and seed curves.

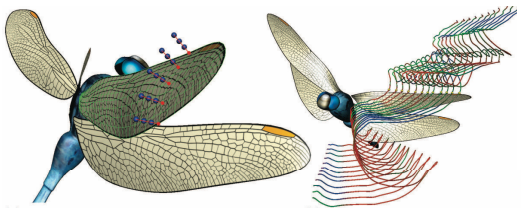

Figure 17: *Placement of seeds (left-hand side image) and the corresponding generated seed curves (right-hand side image) [KWDG11]. Image courtesy of Koehler et al.*

To understand and predict the development of cerebral aneurysms, Behrendt et al. [BBB*18] proposed an interactive flow visualization technique to isolate pathlines near vessel surfaces. After domain experts select patches of the surface that match certain features of interest, a pre-computed set of pathlines is filtered in order to retain pathlines within a threshold distance of the surface patch. The resultant pathline bundle then provides the user with a visualization of the local blood flow pattern. Further, the user can specify multiple patches and use color-coding to differentiate between pathline bundles.

6. Research Challenges

The majority of automated flow visualization algorithms using integral curves operate under the post hoc visualization paradigm and consider a steady state field. As such, there are many potential solutions (SPSS techniques) available. As of writing this, the most obvious opportunities to improve the existing work are in reducing computation. That said, the application of SPSS techniques to unsteady vector fields, in order to visualize integral curves such as pathlines or streaklines in volume flow, still has unsolved problems and continuing challenges. This task is challenging because, unlike streamlines, integral curves in evolving vector fields can intersect in space at different points in time and the amount of vector data to be processed is greater. Further, particles in an unsteady vector field can cluster in certain regions while leaving other regions void. This necessitates tracking of the particle distribution over time to ensure continued coverage of the spatial domain.

In the past decade, multiple works have researched seed placement and stream surfaces selection techniques for the automatic generation of stream surface-based flow visualizations [EML*11, ELM*12, ELC*12, ESRT13, ELM*14, SEG*14, BKC15, BH15,

Technique Target/Context	Dims	State	View-Dependent	Distribution	References
Planar Surface Flow	2D	Steady	No	Uniform	[JL97, JL01, MAD05, ZSW10, ZZS11, ZWZ*13] [VKP00, CML*07, ZD09, WLZMI10]
	2D	Steady	No	Non-Uniform	[LHS08, ZWL15]
	2D	Steady	Yes	Uniform	[TB96]
	2D	Steady	Yes	Non-Uniform	[LSW12]
	2D	Unsteady	No	Uniform	[JL00] [DZC*12]
Curved Surface Flow	3D	Steady	No	Uniform	[MCG94, RPP*09]
	3D	Steady	Yes	Uniform	[MHHI98, SLCZ09] [LM08]
Volume Flow	3D	Steady	No	Uniform	[MTHG03, ZNZ*14] [YKP05, LMG06, LMI07] [CCK07]
	3D	Steady	Yes	Uniform	[LS07, MCHM10, GBWT11, GRT13, MWS13] [LMSC11] [KFW16]
	3D	Steady	Yes	Non-Uniform	[ATR*08] [FI08] [WWYM10]
	3D	Steady	No	Non-Uniform	[ZSH96, Tre00, SHH*07, XLS10, WZN11, YWSC12, MWW*14, SVW16] [TMWS13, LCL*13, TWS14, LWS14, LWS15, HTZ*19, HTW18] [Lar02, LWSH04, LGD*05, PGN*13]
	3D	Unsteady	No	Non-Uniform	[WS05, ZCL*16] [MJL*13] [BJS*98, SRBE99, KWDG11, BBB*18]
	3D	Unsteady	Yes	Uniform	[GRT14]
Streamsurface Construction	3D	Steady	No	Uniform	[Hul92]
Particle-Based Vis	3D	Unsteady	No	Uniform	[BPSS02]
	3D	Steady	No	Non-Uniform	[ELPH18]
Texture-Based Vis	3D	Steady	No	Uniform	[GBH03]
	3D	Unsteady	No	Uniform	[SBL04, HE06]
	3D	Unsteady	No	Non-Uniform	[GGS02]

Table 3: *Grouping of algorithms based on the application context, dimensions, state of flow, dependency on viewpoint, and distribution. References in the right-most column are color-coded based on the type of technique, i.e., density-based (purple), feature-based (blue), similarity-based (red), and manual (green).*

TW16, TW17, Vas17, HTW18]. For stream surfaces, the placement, separation, and alignment of *seed curves* is of interest to researchers. This is an active area of research with new techniques being developed to use stream surfaces effectively.

Uncertainty visualization is an important emerging area of research. Relevant research in the field of uncertain flow visualization has studied the uncertainty that arises in integral curve visualizations from user-input parameters such as seed positions [MET*15]. In another study, Ferstl et al. [FBW15] convey uncertainty in vector field ensembles using streamline variability plots. Both studies rely on identifying similarity between streamlines in proximity, i.e., they use similarity-based methods, and are applied to steady and unsteady vector fields. Future research could utilize these methods for other integral curves (e.g., streaklines) or to identify optimal instances in time to introduce seeds in the domain.

Another area for the application of SPSS techniques is in situ visualization [CBGH19]. In situ processing is typically performed with a limited computational budget and in a distributed memory environment. Initial studies in this space have used SPSS techniques to represent and store steady state vector field data in the form of streamlines [HTZ*19] and unsteady state vector field data in the form of pathlines [ACG*14, BJ15, COJ15, SBC18, SCB19, RPD19]. These methods use SPSS techniques to perform an intelligent sampling and reduction of large vector fields such that they can be accurately reconstructed and explored post hoc. Further, the use of SPSS techniques under in situ computational constraints for the purposes of in situ flow visualization, i.e., generating useful flow visualizations as a large-scale simulation progresses, is relatively unexplored.

7. Conclusion

This report describes the state-of-the-art techniques for seed placement and streamline selection used for flow visualization. The extensive use of streamlines for flow visualization has resulted in several methods and suggested approaches regarding how to use them to explore a flow field. Our classification of these algorithms resulted in three strategy classes (density-based, feature-based, similarity-based) for automatic techniques and two strategy classes (interactive tools, domain information) for manual techniques.

Our survey evaluated automatic techniques to compare and relate them along three axes, namely, redundancy, regions of interest and computation (Figure 2). The three automatic technique classes each offer different benefits. First, density-based techniques provide coverage of the field in either object space or image space and are relatively straightforward. Second, feature-based techniques focus on highlighting salient features of the flow and can often be computed fast. Third, similarity-based techniques minimized redundancy and picked streamlines that together provided the best representative views of the flow. Depending on the use case, different algorithms can be explored or the benefits of different algorithms can be combined (Table 4 summarizes the highlights and potential shortcomings). Further, Table 3 in our survey shows a grouping of SPSS works based on the context, dimension, and state of flow to which a technique is applied. Although multiple different streamline-based flow visualization tasks have been tackled by SPSS techniques, future research applying these techniques to unsteady state flow, stream/path surface visualization, uncertainty flow visualization, and within an in situ flow visualization context is rich in challenging open problems.

Categories of SPSS Techniques	Highlight	Potential Shortcoming
View-Independent or Object Space	Complete domain coverage.	Often contain redundant streamlines.
View-Dependent or Image Space	Efficient occlusion management, prioritization of interesting streamlines for a given view.	Can contain redundant streamlines.
Flow Topology	Great at emphasizing features.	Can contain redundant streamlines and hard to extend some to 3D.
Derived Field	Good at capturing ROI, fast computation.	Can contain redundant streamlines.
User-Defined Scalar Fields	Can be good at capturing ROI, fast computation.	Not guaranteed to capture ROI.
Spatial Distance	Great at controlling redundancy and capturing ROI.	Large number of distance computations and limited to similarity in proximity.
Shape	Can measure similarity in an orientation-, position-, and scale-invariant manner.	Large preprocessing time for large number of streamlines.
Machine Learning	Great at accurately capturing vector field information in the form of streamlines.	Slow to compute.

Table 4: *A summary of the highlights and potential shortcomings of the various automatic SPSS technique categories. Each category name is colored based on its classification, i.e., purple for density-based, blue for feature-based, and red for similarity-based techniques.*

References

[ACG*14] AGRANOVSKY A., CAMP D., GARTH C., BETHEL E. W., JOY K. I., CHILDS H.: Improved post hoc flow analysis via lagrangian representations. In *2014 IEEE 4th Symposium on Large Data Analysis and Visualization (LDAV)* (2014), pp. 67–75. 20

[AMR12] ABRAHAM R., MARSDEN J. E., RATIU T.: *Manifolds, tensor analysis, and applications*, vol. 75. Springer Science & Business Media, 2012. 12

[ATR*08] ANNEN T., THEISEL H., RÖSSL C., ZIEGLER G., SEIDEL H.-P.: Vector field contours. In *Proceedings of Graphics Interface 2008* (2008), Canadian Information Processing Society, pp. 97–105. 8, 20

[BBB*18] BEHRENDT B., BERG P., BEUING O., PREIM B., SAALFELD S.: Explorative blood flow visualization using dynamic line filtering based on surface features. In *Computer Graphics Forum* (2018), vol. 37, Wiley Online Library, pp. 183–194. 19, 20

[BC94] BERNDT D. J., CLIFFORD J.: Using dynamic time warping to find patterns in time series. In *KDD workshop* (1994), vol. 10, Seattle, WA, pp. 359–370. 16

[BH15] BRAMBILLA A., HAUSER H.: Expressive seeding of multiple stream surfaces for interactive flow exploration. *Computers & Graphics 47* (2015), 123–134. 19

[BH17] BUJACK R., HAGEN H.: Moment invariants for multi-dimensional data. In *Modeling, Analysis, and Visualization of Anisotropy*. Springer, 2017, pp. 43–64. 14

[BJ15] BUJACK R., JOY K. I.: Lagrangian representations of flow fields with parameter curves. In *2015 IEEE 5th Symposium on Large Data Analysis and Visualization (LDAV)* (2015), pp. 41–48. 20

[BJS*98] BRYSON S., JOHAN S., SCHLECHT L., GREEN B., KENWRIGHT D., GERALD-YAMASAKI M.: The virtual windtunnel. In *Computational Fluid Dynamics Review 1998: (In 2 Volumes)*. World Scientific, 1998, pp. 1113–1130. 18, 19, 20

[BKC15] BARTOŇ M., KOSINKA J., CALO V. M.: Stretch-minimising stream surfaces. *Graphical Models 79* (2015), 12–22. 19

[BKH*15] BUJACK R., KASTEN J., HOTZ I., SCHEUERMANN G., HITZER E.: Moment invariants for 3d flow fields via normalization. In *2015 IEEE Pacific visualization symposium (PacificVis)* (2015), IEEE, pp. 9–16. 14

[BKKW08] BURGER K., KONDRATIEVA P., KRUGER J., WESTERMANN R.: Importance-driven particle techniques for flow visualization. In *Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific* (2008), Citeseer, pp. 71–78. 13

[BKP*04] BRUN A., KNUTSSON H., PARK H.-J., SHENTON M. E., WESTIN C.-F.: Clustering fiber traces using normalized cuts. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2004), Springer, pp. 368–375. 14

[BPSS02] BAUER D., PEIKERT R., SATO M., SICK M.: A case study in selective visualization of unsteady 3d flow. In *Proceedings of the conference on Visualization'02* (2002), IEEE Computer Society, pp. 525–528. 13, 19, 20

[CBGH19] CHILDS H., BENNETT J., GARTH C., HENTSCHEL B.: In Situ Visualization for Computational Science. *IEEE Computer Graphics and Applications (CG&A) 39*, 6 (Nov./Dec. 2019), 76–85. 20

[CCK07] CHEN Y., COHEN J., KROLIK J.: Similarity-guided streamline placement with error evaluation. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1448–1455. 3, 14, 17, 20

[Che93] CHEW L. P.: Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the ninth annual symposium on Computational geometry* (1993), ACM, pp. 274–280. 6

[CK90] CASH J. R., KARP A. H.: A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides. *ACM Transactions on Mathematical Software (TOMS) 16*, 3 (1990), 201–222. 1

[CLSW14] CHAUDHURI A., LEE T.-Y., SHEN H.-W., WENGER R.: Exploring flow fields using space-filling analysis of streamlines. *IEEE Transactions on Visualization and Computer Graphics 20*, 10 (2014), 1392–1404. 14

[CML*07] CHEN G., MISCHAIKOW K., LARAMEE R. S., PILARCZYK P., ZHANG E.: Vector field editing and periodic orbit extraction using morse decomposition. *IEEE Transactions on Visualization & Computer Graphics*, 4 (2007), 769–785. 10, 11, 20

[Cod12] CODDINGTON E. A.: *An introduction to ordinary differential equations*. Courier Corporation, 2012. 2

[COJ15] CHANDLER J., OBERMAIER H., JOY K. I.: Interpolation-based pathline tracing in particle-based flow visualization. *IEEE Transactions on Visualization and Computer Graphics 21*, 1 (2015), 68–80. 20

[Coo86] COOK R. L.: Stochastic sampling in computer graphics. *ACM Transactions on Graphics (TOG) 5*, 1 (1986), 51–72. 10

[CYY*11] CHEN C.-K., YAN S., YU H., MAX N., MA K.-L.: An illustrative visualization framework for 3d vector fields. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1941–1951. 17

[DZC*12] DING Z., ZHANG X., CHEN W., TRICOCHE X., PENG D., PENG Q.: Coherent streamline generation for 2-d vector fields. *Tsinghua Science and Technology 17*, 4 (2012), 463–470. 10, 20

[EG01] EDELSBRUNNER H., GUOY D.: Sink-insertion for mesh improvement. In *Proceedings of the seventeenth annual symposium on Computational geometry* (2001), ACM, pp. 115–123. 6

[EKS*96] ESTER M., KRIEGEL H.-P., SANDER J., XU X., ET AL.: A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd* (1996), vol. 96, pp. 226–231. 18

[ELC*12] EDMUNDS M., LARAMEE R., CHEN G., ZHANG E., MAX N.: Advanced, automatic stream surface seeding and filtering. *Theory and Practice of Computer Graphics 2012, TPCG 2012 - Eurographics UK Chapter Proceedings* (01 2012), 53–60. 19

[ELM*12] EDMUNDS M., LARAMEE R. S., MALKI R., MASTERS I., CROFT T., CHEN G., ZHANG E.: Automatic stream surface seeding: A feature centered approach. In *Computer Graphics Forum* (2012), vol. 31, Wiley Online Library, pp. 1095–1104. 19

[ELM*14] EDMUNDS M., LARAMEE R. S., MALKI R., MASTERS I., WANG Y., CHEN G., ZHANG E., MAX N.: Interactive stream surface placement a hybrid clustering approach supported by tree maps. In *2014 International Conference on Information Visualization Theory and Applications (IVAPP)* (2014), pp. 347–355. 19

[ELPH18] ENGELKE W., LAWONN K., PREIM B., HOTZ I.: Autonomous particles for interactive flow visualization. In *Computer Graphics Forum* (2018), Wiley Online Library. 13, 20

[ELPZ97] ELDAR Y., LINDENBAUM M., PORAT M., ZEEVI Y. Y.: The farthest point strategy for progressive image sampling. *IEEE Transactions on Image Processing 6*, 9 (1997), 1305–1315. 6

[EML*11] EDMUNDS M., MCLOUGHLIN T., LARAMEE R. S., CHEN G., ZHANG E., MAX N.: Automatic stream surface seeding. In *Eurographics (Short Papers)* (2011), pp. 53–56. 19

[ESRT13] ESTURO J. M., SCHULZE M., RÖSSL C., THEISEL H.: Global selection of stream surfaces. In *Computer Graphics Forum* (2013), vol. 32, Wiley Online Library, pp. 113–122. 19

[FBW15] FERSTL F., BÜRGER K., WESTERMANN R.: Streamline variability plots for characterizing the uncertainty in vector field ensembles. *IEEE Transactions on Visualization and Computer Graphics 22*, 1 (2015), 767–776. 20

[FD07] FREY B. J., DUECK D.: Clustering by passing messages between data points. *Science 315*, 5814 (2007), 972–976. 17

[FG98] FUHRMANN A., GRÖLLER E.: Real-time techniques for 3d flow visualization. In *Proceedings of the conference on Visualization'98* (1998), IEEE Computer Society Press, pp. 305–312. 13

[FI08] FURUYA S., ITOH T.: A streamline selection technique for integrated scalar and vector visualization. In *In IEEE Visualization, Poster Session* (2008). 11, 20

[GBH03] GUO-SHI LI, BORDOLOI U. D., HAN-WEI SHEN: Chameleon: an interactive texture-based rendering framework for visualizing three-dimensional vector fields. In *IEEE Visualization, 2003. VIS 2003.* (2003), pp. 241–248. 5, 20

[GBWT11] GÜNTHER T., BÜRGER K., WESTERMANN R., THEISEL H.: A view-dependent and inter-frame coherent visualization of integral lines using screen contribution. In *VMV* (2011), pp. 215–222. 2, 8, 20

[GGS02] GUTHE S., GUMHOLD S., STRASSER W.: Interactive visualization of volumetric vector fields using texture based particles. In *Journal of WSCG* (2002), pp. 33–41. 13, 20

[GRT13] GÜNTHER T., RÖSSL C., THEISEL H.: Opacity optimization for 3d line fields. *ACM Transactions on Graphics (TOG) 32*, 4 (2013), 120. 3, 8, 16, 20

[GRT14] GÜNTHER T., RÖSSL C., THEISEL H.: Hierarchical opacity optimization for sets of 3d line fields. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 507–516. 8, 9, 20

[HA04] HELGELAND A., ANDREASSEN O.: Visualization of vector fields using seed lic and volume rendering. *IEEE Transactions on Visualization & Computer Graphics*, 6 (2004), 673–682. 6

[HE06] HELGELAND A., ELBOTH T.: High-quality and interactive animations of 3d time-varying vector fields. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1535–1546. 5, 20

[HTW18] HAN J., TAO J., WANG C.: Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces. *IEEE Transactions on Visualization and Computer Graphics* (2018). 4, 18, 19, 20

[HTZ*19] HAN J., TAO J., ZHENG H., GUO H., CHEN D. Z., WANG C.: Flow field reduction via reconstructing vector data from 3-d streamlines using deep learning. *IEEE Computer Graphics and Applications 39*, 4 (2019), 54–67. 15, 20

[Hul92] HULTQUIST J. P. M.: Constructing stream surfaces in steady 3d vector fields. In *Proceedings Visualization '92* (Oct 1992), pp. 171–178. 4, 12, 20

[JH95] JEONG J., HUSSAIN F.: On the identification of a vortex. *Journal of fluid mechanics 285* (1995), 69–94. 13

[JL97] JOBARD B., LEFER W.: Creating evenly-spaced streamlines of arbitrary density. In *Visualization in Scientific Computing '97* (Vienna, 1997), Springer Vienna, pp. 43–55. 5, 6, 7, 10, 11, 12, 20

[JL00] JOBARD B., LEFER W.: Unsteady flow visualization by animating evenly-spaced streamlines. In *Computer Graphics Forum* (2000), vol. 19, Wiley Online Library, pp. 31–39. 5, 20

[JL01] JOBARD B., LEFER W.: Multiresolution flow visualization. In *WSCG* (2001). 5, 20

[KFW16] KANZLER M., FERSTL F., WESTERMANN R.: Line density control in screen-space via balanced line hierarchies. *Computers & Graphics 61* (2016), 29–39. 2, 4, 15, 16, 20

[KKKW05] KRUGER J., KIPFER P., KONCLRATIEVA P., WESTERMANN R.: A particle system for interactive visualization of 3d flows. *IEEE Transactions on Visualization and Computer Graphics 11*, 6 (2005), 744–756. 13

[KW10] KHOURY M., WENGER R.: On the fractal dimension of isosurfaces. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1198–1205. 14

[KWDG11] KOEHLER C., WISCHGOLL T., DONG H., GASTON Z.: Vortex visualization in ultra low reynolds number insect flight. *IEEE Transactions on Visualization & Computer Graphics*, 12 (2011), 2071–2079. 19, 20

[Lar02] LARAMEE R. S.: Interactive 3d flow visualization using a streamrunner. In *Conference on Human Factors in Computing Systems: CHI'02 extended abstracts on Human factors in computing systems* (2002), vol. 20, pp. 804–805. 19, 20

[LCL*13] LU K., CHAUDHURI A., LEE T., SHEN H., WONG P. C.: Exploring vector fields with distribution-based streamline analysis. In *2013 IEEE Pacific Visualization Symposium (PacificVis)* (2013), pp. 257–264. 17, 20

[LGD*05] LARAMEE R. S., GARTH C., DOLEISCH H., SCHNEIDER J., HAUSER H., HAGEN H.: Visual analysis and exploration of fluid flow in a cooling jacket. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 623–630. 19, 20

[LHD*04] LARAMEE R. S., HAUSER H., DOLEISCH H., VROLIJK B., POST F. H., WEISKOPF D.: The state of the art in flow visualization: Dense and texture-based techniques. In *Computer Graphics Forum* (2004), vol. 23, Wiley Online Library, pp. 203–221. 2

[LHS08] LI L., HSIEH H., SHEN H.: Illustrative streamline placement and visualization. In *2008 IEEE Pacific Visualization Symposium* (2008), pp. 79–86. 2, 3, 13, 15, 20

[LHZP07] LARAMEE R. S., HAUSER H., ZHAO L., POST F. H.: Topology-based flow visualization, the state of the art. In *Topology-based methods in visualization*. Springer, 2007, pp. 1–19. 2

[LM08] LIU Z., MOORHEAD R. J.: Interactive view-driven evenly spaced streamline placement. In *Visualization and Data Analysis 2008* (2008), vol. 6809, International Society for Optics and Photonics, p. 68090A. 10, 20

[LMG06] LIU Z., MOORHEAD R., GRONER J.: An advanced evenly-spaced streamline placement algorithm. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 965–972. 10, 11, 20

[LMI07] LIU Z., MOORHEAD II R. J.: Robust loop detection for interactively placing evenly spaced streamlines. *Computing in Science & Engineering 9*, 4 (2007), 86–91. 10, 20

[LMSC11] LEE T.-Y., MISHCHENKO O., SHEN H.-W., CRAWFIS R.: View point evaluation and streamline filtering for flow visualization. In *Visualization Symposium (PacificVis), 2011 IEEE Pacific* (2011), IEEE, pp. 83–90. 12, 20

[LS07] LI L., SHEN H.-W.: Image-based streamline generation and rendering. *IEEE Transactions on Visualization & Computer Graphics*, 3 (2007), 630–640. 7, 20

[LSW12] LUO C., SAFA I., WANG Y.: Feature-aware streamline generation of planar vector fields via topological methods. *Computers & Graphics 36*, 6 (2012), 754–766. 12, 20

[LWS14] LI Y., WANG C., SHENE C.-K.: Streamline similarity analysis using bag-of-features. In *Visualization and Data Analysis 2014* (2014), vol. 9017, International Society for Optics and Photonics, p. 90170N. 17, 20

[LWS15] LI Y., WANG C., SHENE C.-K.: Extracting flow features via supervised streamline segmentation. *Computers & Graphics 52* (2015), 79–92. 18, 20

[LWSH04] LARAMEE R. S., WEISKOPF D., SCHNEIDER J., HAUSER H.: Investigating swirl and tumble flow with a comparison of visualization techniques. In *Visualization, 2004. IEEE* (2004), IEEE, pp. 51–58. 18, 19, 20

[MAD05] MEBARKI A., ALLIEZ P., DEVILLERS O.: Farthest point seeding for efficient placement of streamlines. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 479–486. 6, 10, 11, 12, 20

[MBS*04] MAHROUS K., BENNETT J., SCHEUERMANN G., HAMANN B., JOY K. I.: Topological segmentation in three-dimensional vector fields. *IEEE Transactions on Visualization & Computer Graphics*, 2 (2004), 198–205. 10

[MCG94] MAX N., CRAWFIS R., GRANT C.: Visualizing 3d velocity fields near contour surfaces. In *Proceedings of the conference on Visualization'94* (1994), IEEE Computer Society Press, pp. 248–255. 5, 20

[MCHM10] MARCHESIN S., CHEN C.-K., HO C., MA K.-L.: View-dependent streamlines for 3d vector fields. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1578–1586. 2, 3, 8, 9, 12, 15, 20

[MET*15] MCLOUGHLIN T., EDMUNDS M., TONG C., LARAMEE R. S., MASTERS I., CHEN G., MAX N., YEH H., ZHANG E.: Visualization of input parameters for stream and pathline seeding. *Int. J. Adv. Comput. Sci. Appl.(IJACSA) 6*, 4 (2015), 124–135. 20

[MGWW08] MADDAH M., GRIMSON W. E. L., WARFIELD S. K., WELLS W. M.: A unified framework for clustering and quantitative analysis of white matter fiber tracts. *Medical image analysis 12*, 2 (2008), 191–202. 14

[MH08] MAATEN L. V. D., HINTON G.: Visualizing data using t-sne. *Journal of machine learning research 9*, Nov (2008), 2579–2605. 18

[MHHI98] MAO X., HATANAKA Y., HIGASHIDA H., IMAMIYA A.: Image-guided streamline placement on curvilinear grid surfaces. In *Visualization'98. Proceedings* (1998), IEEE, pp. 135–142. 7, 20

[MJL*13] MCLOUGHLIN T., JONES M. W., LARAMEE R. S., MALKI R., MASTERS I., HANSEN C. D.: Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics 19*, 8 (2013), 1342–1353. 16, 17, 20

[MLP*10] MCLOUGHLIN T., LARAMEE R. S., PEIKERT R., POST F. H., CHEN M.: Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum 29*, 6 (2010), 1807–1829. 1

[MSE*05] MERHOF D., SONNTAG M., ENDERS F., HASTREITER P., FAHLBUSCH R., NIMSKY C., GREINER G.: Visualization of diffusion tensor data using evenly spaced streamlines. *Vision, Modeling and Visualization. Berlin: Akademische Verl.-Ges. AKA, 2005, pp. 257-264* (01 2005). 5

[MTHG03] MATTAUSCH O., THEUSSL T., HAUSER H., GRÖLLER E.: Strategies for interactive exploration of 3d flow using evenly-spaced illuminated streamlines. In *Proceedings of the 19th spring conference on Computer graphics* (2003), ACM, pp. 213–222. 5, 20

[MVVW05] MOBERTS B., VILANOVA A., VAN WIJK J. J.: Evaluation of fiber clustering methods for diffusion tensor imaging. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 65–72. 14, 15

[MWS13] MA J., WANG C., SHENE C.-K.: Coherent view-dependent streamline selection for importance-driven flow visualization. In *Visualization and Data Analysis 2013* (2013), vol. 8654, International Society for Optics and Photonics, p. 865407. 9, 20

[MWW*14] MA J., WALKER J., WANG C., KUHL S., SHENE C. K.: Flowtour: An automatic guide for exploring internal flow features. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific* (2014), IEEE, pp. 25–32. 2, 12, 20

[OB03] OUDOT S., BOISSONNAT J.-D.: Provably good surface sampling and approximation. In *Symposium on Geometry Processing* (2003), pp. 9–18. 6

[OLK*14] OELTZE S., LEHMANN D. J., KUHN A., JANIGA G., THEISEL H., PREIM B.: Blood flow clustering and applications invirtual stenting of intracranial aneurysms. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (2014), 686–701. 14

[OW05] OÂĂŹDONNELL L., WESTIN C.-F.: White matter tract clustering and correspondence in populations. In *International Conference on Medical Image Computing and Computer-Assisted Intervention* (2005), Springer, pp. 140–147. 14

[Pea00] PEARSON K.: X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 50*, 302 (1900), 157–175. 16

[PGN*13] PENG Z., GENG Z., NICHOLAS M., LARAMEE R. S., CROFT N., MALKI R., MASTERS I., HANSEN C.: Visualization of flow past a marine turbine: the information-assisted search for sustainable energy. *Computing and Visualization in Science 16*, 3 (2013), 89–103. 19, 20

[PPF*11] POBITZER A., PEIKERT R., FUCHS R., SCHINDLER B., KUHN A., THEISEL H., MATKOVIĆ K., HAUSER H.: The state of the art in topology-based visualization of unsteady flow. In *Computer Graphics Forum* (2011), vol. 30, Wiley Online Library, pp. 1789–1811. 2

[PPG12] PUGMIRE D., PETERKA T., GARTH C.: Parallel integral curves. *High Performance Visualization: Enabling Extreme-Scale Scientific Insight* (2012), 91–113. 1

[PVH*03] POST F. H., VROLIJK B., HAUSER H., LARAMEE R. S., DOLEISCH H.: The state of the art in flow visualisation: Feature extraction and tracking. In *Computer Graphics Forum* (2003), vol. 22, Wiley Online Library, pp. 775–792. 2

[PYK*18] PUGMIRE D., YENPURE A., KIM M., KRESS J., MAYNARD R., CHILDS H., HENTSCHEL B.: Performance-Portable Particle Advection with VTK-m. In *Eurographics Symposium on Parallel Graphics and Visualization* (2018), The Eurographics Association. 1

[RPD19] RAPP T., PETERS C., DACHSBACHER C.: Void-and-cluster sampling of large scattered data and trajectories. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2019), 780–789. 20

[RPP*09] ROSANWO O., PETZ C., PROHASKA S., HEGE H.-C., HOTZ I.: Dual streamline seeding. In *Visualization Symposium, 2009. PacificVis' 09. IEEE Pacific* (2009), IEEE, pp. 9–16. 6, 20

[SBC18] SANE S., BUJACK R., CHILDS H.: Revisiting the Evaluation of In Situ Lagrangian Analysis. In *Eurographics Symposium on Parallel Graphics and Visualization* (2018), The Eurographics Association. 20

[SBL04] SHEN H.-W., BORDOLOI U. D., LI G.-S.: Interactive visualization of three-dimensional vector fields with flexible appearance control. *IEEE Transactions on Visualization and Computer Graphics 10*, 4 (2004), 434–445. 5, 20

[SC17] SHI L., CHEN G.: Metric-based curve clustering and feature extraction in flow visualization. 14

[SCB19] SANE S., CHILDS H., BUJACK R.: An Interpolation Scheme for VDVP Lagrangian Basis Flows. In *Eurographics Symposium on Parallel Graphics and Visualization* (2019), The Eurographics Association. 20

[SEG*14] SCHULZE M., ESTURO J. M., GÜNTHER T., RÖSSL C., SEIDEL H.-P., WEINKAUF T., THEISEL H.: Sets of globally optimal stream surfaces for flow visualization. In *Computer Graphics Forum* (2014), vol. 33, Wiley Online Library, pp. 1–10. 19

[SGSM08] SALZBRUNN T., GARTH C., SCHEUERMANN G., MEYER J.: Pathline predicates and unsteady flow structures. *The Visual Computer 24*, 12 (2008), 1039–1051. 14

[SH95a] STALLING D., HEGE H.-C.: Fast and resolution independent line integral convolution. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques* (1995), ACM, pp. 249–256. 13

[SH95b] SUJUDI D., HAIMES R.: Identification of swirling flow in 3-d vector fields. In *12th Computational Fluid Dynamics Conference* (1995), p. 1715. 19

[SHH*07] SCHLEMMER M., HOTZ I., HAMANN B., MORR F., HAGEN H.: Priority streamlines: A context-based visualization of flow fields. In *EuroVis* (2007), pp. 227–234. 14, 20

[SLC19] SHI L., LARAMEE R. S., CHEN G.: Integral curve clustering and simplification for flow visualization: A comparative evaluation. *IEEE Transactions on Visualization and Computer Graphics* (2019). 14

[SLCZ09] SPENCER B., LARAMEE R. S., CHEN G., ZHANG E.: Evenly spaced streamlines for surfaces: An image-based approach. In *Computer Graphics Forum* (2009), vol. 28, Wiley Online Library, pp. 1618–1631. 7, 20

[SRBE99] SCHULZ M., RECK F., BARTELHEIMER W., ERTL T.: Interactive visualization of fluid dynamics simulations in locally refined cartesian grids (case study). In *Proceedings of the conference on Visualization'99: celebrating ten years* (1999), IEEE Computer Society Press, pp. 413–416. 19, 20

[SS06] SALZBRUNN T., SCHEUERMANN G.: Streamline predicates. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1601–1612. 14

[SVW16] SHEN H.-W., VASKO R., WENGER R.: Visualizing flow fields using fractal dimensions. In *Proceedings of the Eurographics/IEEE VGTC Conference on Visualization: Short Papers* (2016), Eurographics Association, pp. 25–29. 14, 20

[TB96] TURK G., BANKS D.: Image-guided streamline placement. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques* (1996), ACM, pp. 453–460. 5, 6, 7, 9, 10, 20

[TMWS13] TAO J., MA J., WANG C., SHENE C.-K.: A unified approach to streamline selection and viewpoint selection for 3d flow visualization. *IEEE Transactions on Visualization and Computer Graphics 19*, 3 (2013), 393–406. 15, 18, 20

[Tre00] TREINISH L. A.: Multi-resolution visualization techniques for nested weather models. In *Proceedings of the conference on Visualization'00* (2000), IEEE Computer Society Press, pp. 513–516. 9, 20

[TW16] TAO J., WANG C.: Peeling the flow: A sketch-based interface to generate stream surfaces. In *SIGGRAPH ASIA 2016 Symposium on Visualization* (2016), ACM, p. 14. 19

[TW17] TAO J., WANG C.: Semi-automatic generation of stream surfaces via sketching. *IEEE Transactions on Visualization and Computer Graphics* (2017). 19

[TWHS03] THEISEL H., WEINKAUF T., HEGE H.-C., SEIDEL H.-P.: Saddle connectors-an approach to visualizing the topological skeleton of complex 3d vector fields. In *Visualization, 2003. VIS 2003. IEEE* (2003), IEEE, pp. 225–232. 10

[TWHW07] TSAI A., WESTIN C.-F., HERO A. O., WILLSKY A. S.: Fiber tract clustering on manifolds with dual rooted-graphs. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on* (2007), IEEE, pp. 1–6. 14

[TWS14] TAO J., WANG C., SHENE C. K.: Flowstring: Partial streamline matching using shape invariant similarity measure for exploratory flow visualization. In *Visualization Symposium (PacificVis), 2014 IEEE Pacific* (2014), IEEE, pp. 9–16. 17, 20

[Vas17] VASKO R.: *Techniques for Assistance in Streamline and Stream Surface Visualizations*. PhD thesis, The Ohio State University, 2017. 19

[VBVP04] VILANOVA A., BERENSCHOT G., VAN PUL C.: Dti visualization with streamsurfaces and evenly-spaced volume seeding. In *Proceedings of the Sixth Joint Eurographics-IEEE TCVG conference on Visualization* (2004), Eurographics Association, pp. 173–182. 5

[VKP00] VERMA V., KAO D., PANG A.: A flow-guided streamline seeding strategy. In *Proceedings of the conference on Visualization'00* (2000), IEEE Computer Society Press, pp. 163–170. 2, 3, 9, 10, 20

[VPBB*11] VAN PELT R., BESCOS J. O., BREEUWER M., CLOUGH R. E., GROLLER M. E., TER HAAR ROMENIJ B., VILANOVA A.: Interactive virtual probing of 4d mri blood-flow. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2153–2162. 13

[WF74] WAGNER R. A., FISCHER M. J.: The string-to-string correction problem. *Journal of the ACM (JACM) 21*, 1 (1974), 168–173. 16

[WHN*03] WEINKAUF T., HEGE H.-C., NOACK B. R., SCHLEGEL M., DILLMANN A.: Coherent structures in a transitional flow around a backward-facing step. *Physics of Fluids 15*, 9 (2003), S3–S3. 13

[WLZMI10] WU K., LIU Z., ZHANG S., MOORHEAD II R. J.: Topology-aware evenly spaced streamline placement. *IEEE Transactions on Visualization and Computer Graphics 16*, 5 (2010), 791–801. 4, 11, 20

[WS05] WIEBEL A., SCHEUERMANN G.: Eyelet particle tracing-steady visualization of unsteady flow. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 607–614. 12, 20

[WT02] WEINKAUF T., THEISEL H.: Curvature measures of 3d vector fields and their applications. *Journal of WSCG 10*, 2 (February 2002), 507–514. 13

[WWYM10] WEI J., WANG C., YU H., MA K.-L.: A sketch-based interface for classifying and visualizing vector fields. In *Visualization Symposium (PacificVis), 2010 IEEE Pacific* (2010), IEEE, pp. 129–136. 16, 20

[WZN11] WANG Y., ZHANG W., NING J.: Streamline-based visualization of 3d explosion fields. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on* (2011), IEEE, pp. 1224–1228. 12, 20

[XLS10] XU L., LEE T.-Y., SHEN H.-W.: An information-theoretic framework for flow visualization. *IEEE Transactions on Visualization and Computer Graphics 16*, 6 (2010), 1216–1224. 11, 12, 13, 15, 18, 20

[YKP05] YE X., KAO D., PANG A.: Strategy for seeding 3d streamlines. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 471–478. 10, 20

[YWSC12] YU H., WANG C., SHENE C.-K., CHEN J. H.: Hierarchical streamline bundles. *IEEE Transactions on Visualization and Computer Graphics 18*, 8 (2012), 1353–1367. 13, 15, 16, 20

[ZCL*16] ZHANG L., CHEN G., LARAMEE R. S., THOMPSON D., SESCU A.: Flow visualization based on a derived rotation field. *Electronic Imaging 2016*, 1 (2016), 1–10. 13, 20

[ZD09] ZHANG W., DENG J.: Topology-driven streamline seeding for 2d vector field visualization. In *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on* (2009), IEEE, pp. 4901–4905. 10, 20

[ZNZ*14] ZHANG W., NING J., ZHANG M., PEI Y., LIU B., SUN B.: Multiresolution streamline placement based on control grids. *Integrated Computer-Aided Engineering 21*, 1 (2014), 47–57. 6, 20

[ZS09] ZHANG W., SU J.: Extraction of limit streamlines in 2d flow field using virtual boundary. In *Computational Intelligence and Security, 2009. CIS'09. International Conference on* (2009), vol. 1, IEEE, pp. 171–175. 11

[ZSH96] ZOCKLER M., STALLING D., HEGE H.-C.: Interactive visualization of 3d-vector fields using illuminated stream lines. In *Visualization'96. Proceedings.* (1996), IEEE, pp. 107–113. 13, 20

[ZSW10] ZHANG W., SUN B., WANG Y.: A streamline placement method highlighting flow field topology. In *Computational Intelligence and Security (CIS), 2010 International Conference on* (2010), IEEE, pp. 238–242. 6, 20

[ZWL15] ZHENG L., WANG W., LI S.: Feature-based streamline selection method for 2d flow fields. In *Computer-Aided Design and Computer Graphics (CAD/Graphics), 2015 14th International Conference on* (2015), IEEE, pp. 129–136. 16, 20

[ZWZ*13] ZHANG W., WANG Y., ZHAN J., LIU B., NING J.: Parallel streamline placement for 2d flow fields. *IEEE Transactions on Visualization and Computer Graphics 19*, 7 (2013), 1185–1198. 6, 20

[ZZS11] ZHANG W., ZHANG M., SUN B.: Multiresolution streamline placement for 2d flow fields. In *Computational Intelligence and Security (CIS), 2011 Seventh International Conference on* (2011), IEEE, pp. 1174–1178. 6, 20