FlowGraph: A Compound Hierarchical Graph for Flow Field Exploration

Jun Ma* Chaoli Wang† Ching-Kuang Shene‡

Michigan Technological University

Abstract

Visual exploration of large and complex 3D flow fields is critically important for understanding many aero- and hydro-dynamical systems that dominate various physical and natural phenomena in the world. In this paper, we introduce the FlowGraph, a novel compound graph representation that organizes streamline clusters and spatial regions hierarchically for occlusion-free and controllable visual exploration. Our approach works with any seeding strategies as long as the domain is well covered and important flow features are captured. By transforming a flow field to a graph representation, we enable observation and exploration of the relationships among streamline clusters, spatial regions and their interconnection in the transformed space. The FlowGraph not only provides a visual mapping that abstracts streamline clusters and spatial regions in various levels of detail, but also serves as a navigation tool that guides flow field exploration and understanding. Through brushing and linking in conjunction with the spatial streamline view, we demonstrate the effectiveness of FlowGraph with several visual exploration and comparison tasks that can not be well accomplished using the streamline view alone. As occlusion and clutter are almost ubiquitous in 3D flows, the FlowGraph represents a promising direction for enhancing our ability to understand large and complex flow field data.

1 Introduction

Flow visualization plays a vital role in many scientific, engineering and medical disciplines, offering users a graphical representation of their vector data for visual understanding, interpretation and decision making. For more than two decades, flow visualization has been a central topic in scientific visualization and a variety of techniques, including glyph-based [17], texture-based [12], integration-based [14], partition-based [19] and illustration-based [1] techniques have been presented. We focus on integration-based flow visualization as it is most widely used in practice. For integration-based flow visualization, particles or seeds are placed in a vector field and advected over time. The traces or field-lines that the particles follow, i.e., streamlines for steady flow and pathlines for unsteady flow, depict the underlying vector data.

The ever-growing size and complexity of flow data produced from scientific simulations pose significant challenges which are not thoroughly addressed by existing visualization techniques. Among them, a fundamental challenge is the poor scaling of visualization algorithms from 2D flow to 3D flow visualization due to occlusion and clutter. When depicting a 3D flow field using streamlines, it is often possible to reduce spatial occlusion (e.g., through streamline seeding or filtering) but not eliminate it. This prevents an occlusion-free observation and comparison of the relationships among streamlines, a critical task commonly found in many flow field applications. This challenge was echoed in recent state of the art reports on flow visualization [14, 1]. Furthermore, even though streamlines can be organized into a hierarchy to facilitate the understanding [9, 23, 27], visual exploration could still remain a significant challenge due to the lack of capability to observe streamlines and their spatial relationships in a controllable fashion.

We therefore propose to design the FlowGraph, a visual representation and an interface for exploring a 3D flow field. The FlowGraph transforms streamline clusters and spatial regions into a compound hierarchical graph representation to support effective relationship overview and detail exploration in conjunction with the spatial view of streamlines. Through brushing and linking, the user can easily make connection between the graph view and the streamline view. We specifically design a set of functions that enable hierarchical exploration of streamline clusters, spatial regions and their interconnection, detail comparison among streamline clusters in terms of their paths passing through different spatial regions, and close examination of spatial regions by comparing different streamline clusters passing through them. Animation is leveraged to help intuitive comprehension of graph transition and path illustration. A graph layout algorithm is realized to maintain stable graph update during the level-of-detail exploration. We also introduce animated transition that switches between the entire compound graph and the streamline cluster or spatial region subgraph, allowing observation of the subgraphs in a less cluttered view. We demonstrate the effectiveness of the FlowGraph with several flow field data sets of various characteristics. Our results show that the FlowGraph can substantially augment our ability to understand and explore a flow field in different levels of detail, providing the clarity and flexibility previously unavailable.

2 Related Work

Flow Field Exploration Techniques. Visual exploration of 3D flow fields remains quite a challenge for which a variety of solutions have been proposed. For instance, Heiberg et al. [10] proposed to locate, identify and visualize a set of predefined structures in 3D flows using vector pattern matching. Schlemmer et al. [20] presented the idea of invariant moments for analyzing 2D flow fields which allows extraction and visualization of 2D flow patterns, invariant under translation, scaling and rotation. Rössl and Theisel [18] mapped streamlines to points based on the preservation of the Hausdorff metric in the streamline space. The image of the set of streamlines covering the vector field is a set of 2-manifolds embedding in $\mathbf{R}^n$ with characteristic geometry and topology. Other researchers investigated sketch-based interface and interaction for intuitive flow field exploration. For example, Schroeder et al. [21] presented a sketch-based interface for illustrative 2D vector field visualization which allows illustrators to draw directly on top of the data. Their interface design strikes a good balance between supporting artistic freedom and maintaining the accuracy with respect to the underlying vector field data. Wei et al. [24] targeted 3D flow fields and presented a solution that allows the user to sketch a 2D curve for pattern matching in 2D and streamline clustering in 3D. They also explored another way that creates streamline templates hierarchically to support on-the-fly partial streamline matching in a progressive manner.

Focus+Context Flow Visualization. To enable greater control of interesting flow features and patterns for detail examina-

*e-mail: junm@mtu.edu

†e-mail: chaoliw@mtu.edu

‡e-mail: shene@mtu.edu

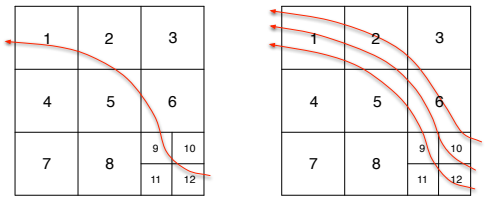


Figure 1: Illustration of L-node signature with a 2D space partitioning example. (a) the signature of the streamline is an ordered sequence $(12,10,9,6,5,2,1)$. (b) the signature of the streamline cluster is an unordered set $(1,2,3,5,6,9,10,11,12)$.

tion, researchers also explored different focus+context techniques. Fuhrmann and Gröller [7] presented magic lenses and magic boxes to examine the region of interest with greater detail by showing denser streamlines. This technique was extended to magic volumes of varying focus regions such as cubes, prisms and spheres [13]. Laramee et al. [11] leveraged feature-based techniques [5] to extract interesting flow regions, such as stagnant flow, reverse-longitudinal flow and regions of high pressure gradient as the focus and achieved focus+context rendering through interactive thresholding. Correa et al. [3] introduced physical and optical operators to intuitively visualize the internal 3D flow through illustrative deformation. By cutting along flow traces, they allowed clear observation of the internal 3D flow through optical transformation and elastic deformation. To explore blood flow in cerebral aneurysms, Gasteiger et al. [8] proposed an interactive 2D widget for flexible visual filtering and visualization of the focus+context pairs (i.e., relevant hemodynamic attributes). Their widget supports local probing and conveys changes over time for the lens region.

Comparison with Flow Web. Closely related to our work is the flow web presented by Xu and Shen [25] for 3D flow field exploration. In their flow web, a node represents a region in the domain and the strength of a link between two nodes indicates the number of particles traveling between the two regions. Similar graph representations have also been employed for workload estimation for parallel and out-of-core streamline generation [16, 2]. Since the flow web does not explicitly store information about streamline clusters, queries such as identifying streamline bundles become a trial-and-error process. It works for structural flow fields where a path going through a list of nodes may indeed indicate streamline passing through the corresponding regions in order. However, for turbulent flow fields, this may not be true anymore. Rather than only considering streamline clusters or spatial regions as nodes, our FlowGraph integrates both streamline clusters or spatial regions as nodes and thus presents a more complete picture. In this regard, the flow web is actually a subgraph of the FlowGraph (without L-nodes, L-L edges and L-R edges). The FlowGraph allows the user to fully explore their relationships through interacting with the graph view and making connection to the streamline view.

3 FlowGraph Definition and Construction

We define the *FlowGraph* as a compound hierarchical graph that consists of two kinds of nodes and three kinds of edges:

- *R-nodes:* A R-node represents a spatial region. We partition the volume space hierarchically using octree and each non-leaf R-node consists of eight child R-nodes. Each R-node maintains three lists recording the streamlines going in, staying inside or going out of the R-node, respectively.
- *L-nodes:* A leaf L-node corresponds to a single streamline, and a non-leaf L-node represents a cluster of streamlines. We organize streamlines hierarchically and each non-leaf L-node usually consists of a different number of child L-nodes. Each L-node maintains a R-node string which indicates the leaf-level regions which the L-node goes through. If the L-node is a single streamline, the string records a *sequence* of the leaf-level regions it traverses in order. Otherwise, this string records a *set* of the leaf-level regions traversed by all streamlines in the L-node without ordering. We call this string the *signature* of the L-node and define the size of the L-node as the size of its signature, i.e., the number of leaf-level regions. Figure 1 illustrates these two kinds of L-node signatures in a 2D scenario.
- *R-R edges:* A R-R edge is formed between two R-nodes at the same level of the space hierarchy. The edge weight records the number of common streamlines shared by these two R-nodes.
- *L-L edges:* A L-L edge is formed between two L-nodes at the same level of the streamline hierarchy. The edge weight records the number of common leaf-level regions traversed in order by these two L-nodes.
- *L-R edges:* A L-R edge is formed between a L-node and a R-node to show their interconnection. The edge weight records the number of streamlines in the L-node passing through the R-node.

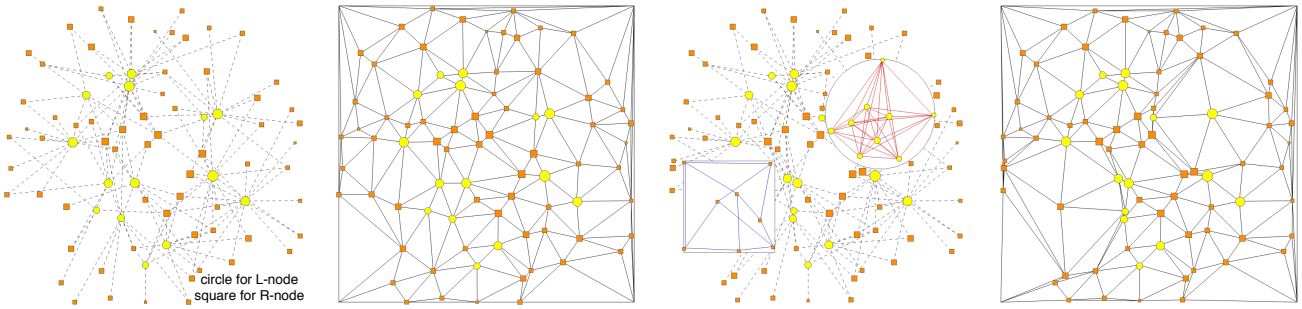
3.1 Space Hierarchy Construction

We form the space hierarchy by partitioning the spatial domain evenly in a top-down manner using octree. Starting from the entire volume as a single region, we compute the flow entropy based on the joint distribution of vector magnitudes and directions for all vectors within. We partition each region further only if its entropy value per voxel is larger than a given threshold. The smallest size of a spatial region is also given as another termination condition. This produces a spatial partition similar to an adaptive mesh refinement (AMR) grid.

3.2 Streamline Hierarchy Construction

To construct the streamline hierarchy, we group spatially neighboring and geometrically similar streamlines in a bottom-up manner. We define the following two types of similarity to measure the distance between streamlines and the distance between streamline clusters, respectively:

- *Streamline similarity* (for leaf level L-nodes): We consider two factors when computing the similarity between two streamlines l_1 and l_2: the *longest common subsequence* (LCS) of the signatures of l_1 and l_2 and the *mean of closest region distances* (MCR) between l_1 and l_2. We define the distance between two regions as the distance of their center points. The MCR is a approximation of the *mean of the closest point distance* (MCP) [15] between two streamlines. Specifically, we treat each streamline as a point sequence which consists of the center points of all leaf regions in the streamline's signature. We compute the MCR of two streamlines as the MCP between their center point sequences. Since the number of regions for a streamline is much smaller than the number of points on the streamline, our MCR has a much lower computation complexity than the MCP does. Furthermore, since the MCR is always computed by using regions at the finest level, its accuracy is also acceptable. The final similarity between two streamlines l_1 and l_2 is defined as
$$\Phi(l_1,l_2) = \frac{\mathrm{LCS}(l_1,l_2)}{\max(|l_1|,|l_2|)} - \frac{\mathrm{MCR}(l_1,l_2)}{\mathrm{MCR}_{\max l}}, \qquad (1)$$
where $\max(|l_1|,|l_2|)$ is the maximum signature size of l_1 and l_2, and $\mathrm{MCR}_{\max l}$ is the maximum MCR among all pairs of streamlines.
- *Streamline cluster similarity* (for non-leaf level L-nodes): Given two streamline clusters c_1 and c_2, we consider two factors for determining their similarity. The MCR is the first factor and we apply the same method used in calculating streamline similarity to the two representative streamlines, one for c_1 and the other for c_2. To determine the spatial overlap of c_1 and



(a) initial layout (b) triangle mesh of (a) (c) adjusted layout (d) triangle mesh of (c)

Figure 2: (a) the initial layout is produced using the force-directed graph layout algorithm. The size of each node in the graph is proportional to the number of children within. (b) the triangle mesh produced from the initial node positions. (c) the adjusted layout after two nodes are selected and expanded for detail examination. (d) the underlying triangle mesh is used to maintain the topology of the graph during layout adjustment.

c_2, we define the second factor as the *shared set* (SS) of the signatures of c_1 and c_2. Unlike the LCS computation which considers the order in the signature, the shared set records all common leaf-level regions shared by the two signatures. Finally, we define the similarity between two streamline clusters c_1 and c_2 as

$$\Phi(c_1,c_2) = \frac{\mathrm{SS}(c_1,c_2)}{\max(|c_1|,|c_2|)} - \frac{\mathrm{MCR}(c_1,c_2)}{\mathrm{MCR}_{\max c}}, \qquad (2)$$

where $\max(|c_1|,|c_2|)$ is the maximum signature size of c_1 and c_2, and $\mathrm{MCR}_{\max c}$ is the maximum MCR among all pairs of streamline clusters.

As we can see, these two similarity definitions are very similar. We replace LCS with SS in the cluster similarity computation. This is because multiple traversal orders may exist for a cluster containing more than one streamline. For the rest of the paper, we do not distinguish these two similarity definitions explicitly and simply state them as the similarity between two L-nodes.

With streamline similarity and streamline cluster similarity defined, we take a bottom-up approach to group streamlines level by level to construct the streamline hierarchy. For each level, we pick the L-node with the *longest* signature size as the first representative and put it into the representative pool. Then, for all other L-nodes, we compute their similarity to the representative pool. We define $\Phi(l,p)$, i.e., the similarity of one L-node to the representative pool, as the *maximum* similarity of this L-node to all representatives currently in the pool, where l denotes the L-node and p denotes the pool. By combining $\Phi(l,p)$ with the L-node signature size $|l|$, we define the representative value of l as

$$\upsilon_l = \left(1 - \frac{\Phi(l,p)}{\max\{\Phi(l,p)\}}\right) + \frac{|l|}{\max\{|l|\}}, \qquad (3)$$

where $\max\{\Phi(l,p)\}$ denotes the maximum $\Phi(l,p)$, and $\max\{|l|\}$ denotes the maximum L-node signature size among all representative candidates. The next representative is the one with the maximum υ_l which means this L-node is not only dissimilar with any representatives in the pool (a low value of $\Phi(l,p)$) but also traverses a relatively long path (a large value of $|l|$). Then we put the new representative into the pool and repeat this process until we identify enough representatives for this level. Now we cluster each of the rest of L-nodes into one of the representatives which this L-node is most similar to. Finally, we obtain a new set of L-node clusters and make it the input set for the next level clustering. We repeat the entire process until a certain number of streamline levels is created.

In practice, for constructing the FlowGraph, it is desirable for spatial regions or streamline clusters to have three to five levels in their respective hierarchy. This is suggested through some empirical measure of the resulting graph complexity. For the streamline hierarchy, the actual number of levels could be larger while we only use several levels at the topmost of the hierarchy for FlowGraph drawing. This would allow us to draw the FlowGraph in an efficient way and maintain a good balance between clarity and complexity.

4 FlowGraph Drawing

We apply the Fruchterman-Reingold algorithm, a classical force-directed graph layout algorithm [6] to draw the compound FlowGraph in 2D. To distinguish among different kinds of nodes, we use nodes of different colors and shapes: orange squares for R-nodes and yellow circles for L-nodes. An example is shown in Figure 2 with the solar plume data set. We also use edges of different colors and styles. In Figure 2, L-R edges are drawn in gray dashed lines. For the underlying graph representation, L-L edges and L-R edges are undirected while R-R edges are directed. Given two regions r_1 and r_2, we differentiate between streamlines going from r_1 to r_2 and streamlines going from r_2 to r_1. For simplicity, instead of using double directed R-R edges, we draw a single undirected R-R edge using the summation of the numbers of streamlines passing through these two regions. While all L-L edges and L-R edges are used for computing the layout, for R-R edges, we only use edges that across *neighboring* spatial regions. This prevents the force model from pulling two R-nodes together although they are far away in the spatial domain. The resulting FlowGraph will better reflect the underlying structural relationships among different R-nodes.

At runtime, the user explores the streamline hierarchy or the space hierarchy by clicking a node in the FlowGraph to expand and examine finer detail. Therefore, we need to adjust the layout to accommodate such level-of-detail explorations. A good layout should maintain a good balance between preserving the structural information of the graph and revealing the dynamics while reducing overlap or occlusion. We generate the initial layout for the coarsest level of the FlowGraph. To achieve stable update, we apply a triangulation scheme [22] to this initial graph and use the result of the triangulation to perform constrained layout adjustment. The four corners of the drawing area are considered as pseudo-nodes in the triangulation. When a node is expanded in the FlowGraph, its initial size is proportional to the number of children in its next level of detail. All nodes expanded are assigned the same scaling factor. The user can also shrink an expanded node back by clicking the empty region inside of the expended node. The surrounding nodes which are pushed away due to the expansion will be pulled back to their respective positions as much as possible.

Similar to the work presented in [4], we consider four kinds of forces to reposition the nodes to reduce their overlap while maintaining the topology of the coarsest level of the FlowGraph. These forces include: a *bidirectional repulsive force* which pushes away two nodes u and v from each other and is effective iff u and v overlap each other, a *unidirectional repulsive force* which pushes away a node u without detail shown from a node v with detail shown and is effective iff u is inside of v, a *spring force* which offsets the

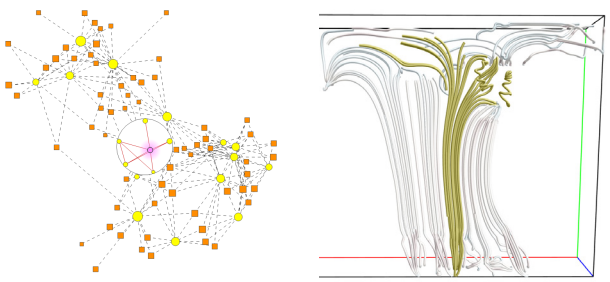


Figure 3: A L-node is expanded in the computer room data set and one of its child node is shown in purple. The corresponding child and parent streamline clusters are shown in gold and white, respectively.

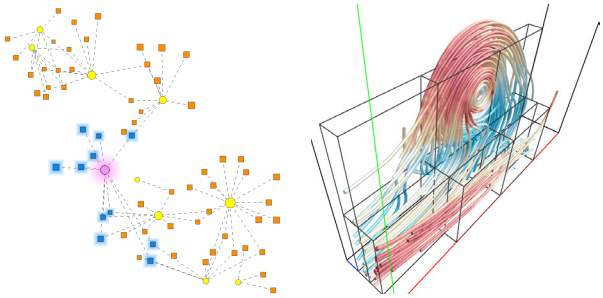


Figure 4: Filtering L-R edges by weight in the FlowGraph highlights eleven R-nodes (shown in blue) that have strong connection with the L-node of interest (shown in purple) in the hurricane data set.

two repulsive forces introduced by reducing the gap between every pair of nodes in the graph, and an *attractive force* which maintains the topology of the underlying triangle mesh by flipping a triangle back if it is flipped. More detail about the computation of these four forces can be found in [4]. Figure 2 shows an example of layout adjustment during the level-of-detail exploration. As we can see, the expanded nodes expel other nodes outside of their regions while the global structure of the FlowGraph is still preserved. We apply this layout adjustment strategy recursively to nodes at different hierarchical levels in the same way.

5 FlowGraph Exploration and Interrogation

The FlowGraph contains a wealth of information that can be effectively utilized for flow field exploration and interrogation. By simply observing the graph, we can already obtain some helpful hints. In a single subgraph (e.g., only R-nodes with R-R edges, or only L-nodes with L-L edges), the size and degree of nodes indicate their importance or significance in the flow field. For instance, if the degree of a R-node is high which means that this R-node has connection to many other R-nodes in terms of streamlines passing through them, it is likely that either this R-node is close to the center of the volume or this R-node contains some critical points such as a sink or source. If the size of a L-node is large, we know that this L-node represents a large streamline cluster. The distance between two nodes also indicates how close their relationship is or how tight their connection is. To extract further information and knowledge about the underlying flow field, we provide the following ways of exploring the graph view and the streamline view.

5.1 Hierarchical Exploration

The FlowGraph organizes L-nodes and R-nodes hierarchically. The user can select a node of interest and expand it to see its next level of detail recursively. In a similar way, the user can further explore each of the nodes at the higher level of detail and make connection to the spatial streamline view. We provide the hierarchical exploration in both the compound graph and a single subgraph. Keyboard shortcuts are added to support convenient traversal through sibling nodes as well as ancestor or descendent nodes.

To provide better context when exploring streamline clusters, we give the option to show the two consecutive levels of streamline clusters in two different colors: the child cluster in a bright color and the rest in a low saturated color. Figure 3 shows such an example. The constrained layout adjustment algorithm (Section 4) guarantees smooth update of the FlowGraph layout when the user explores nodes at various levels of the hierarchy. Similarly, we support the same strategy of hierarchical exploration in the streamline view by allowing the user to visit streamline clusters or spatial regions in various levels of detail.

5.2 Brushing and Linking

Brushing and linking are the standard technique to make connection among multiple views. We dynamically connect the graph view and the streamline view together: when the user clicks a L-node (R-node) in the graph view, its corresponding streamline cluster (spatial region) is highlighted in the streamline view. Conversely, the corresponding L-node (R-node) will be highlighted in the graph view when the user selects a streamline cluster (spatial region) in the streamline view.

To select a streamline cluster, we allow the user to first mouse over the streamlines displayed and click on a streamline of interest. The streamline cluster corresponding to the topmost level node in the FlowGraph will be identified and highlighted. The user can then use the keyboard shortcuts to move up (down) the streamline hierarchy to select a parent (child) node accordingly.

To select a spatial region, we allow the user to first use sliders to identify the ranges in the x, y and z directions, respectively. Mouse-over selection is also enabled for convenience. The spatial region corresponding to the leaf node in the FlowGraph will be identified and highlighted. Moving up or down the space hierarchy is similar to the streamline hierarchy.

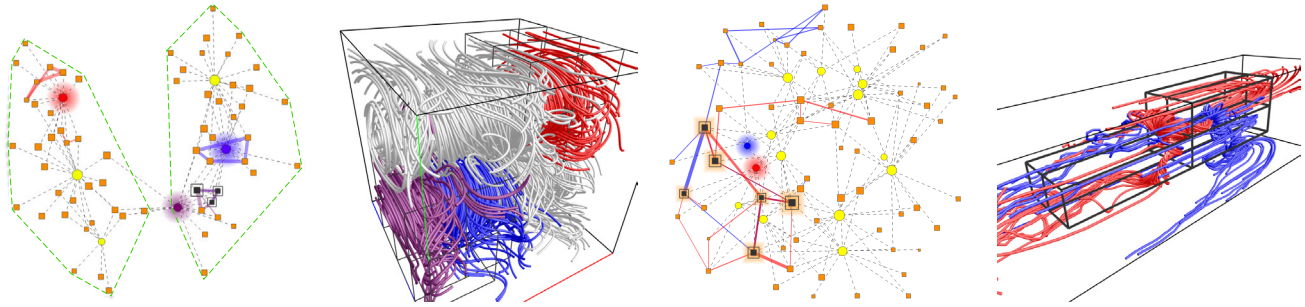
As an option, when a streamline cluster is selected, the corresponding spatial regions which the cluster traverses will be highlighted in the streamline view and at the same time, the corresponding paths passing R-nodes will also be highlighted in the graph view. Similarly, when a spatial region is selected, the corresponding streamline clusters passing through the region will be displayed in the streamline view, and at the same time, the corresponding L-nodes will be highlighted in the graph view. Through brushing and linking, especially combined with hierarchical exploration, the users can quickly build up their mental connection between the intuitive streamline view and the abstract graph view. This will greatly help further exploration which we introduce in the following.

5.3 Filtering and Querying

Given a large and complex 3D flow field, the resulting FlowGraph will consist of a large number of nodes and edges of different kinds. Filtering and querying the graph helps reduce the complexity of both the graph view and the streamline view, allowing the user to focus on the nodes and edges of interest for detail exploration. We provide a set of queries, including node query (by degree or weight) and edge query (by weight), to assist the visual exploration of the FlowGraph.

For FlowGraph nodes, filtering L-nodes by L-L edge weight allows querying streamline clusters based on their path similarity; filtering L-nodes by L-R edge degree allows querying streamline clusters by their spatial extents; filtering R-nodes by R-R edge weight allows querying spatial regions based on their streamline density; and filtering R-nodes by L-R edge degree allows querying spatial regions by their spatial complexity.

For FlowGraph edges, filtering L-L edges by weight allows querying the degree of similarity between streamline clusters in terms of common regions traversed; filtering R-R edges by weight allows querying the strength of connection between spatial regions



(a) (b) (c) (d)

Figure 5: (a) and (b) show path comparison for three streamline clusters shown in red, blue and purple, respectively of the two swirls data set. Observe that the two swirls are well separated in the graph view as indicated by the green dashed lines. Three R-nodes shared by the blue and purple L-nodes are highlighted with double boundaries. (c) and (d) show path comparison for two streamline clusters of the solar plume data set.

in terms of streamline connectivity; and filtering L-R edges by weight allows querying the strength of interconnection between streamline clusters and spatial regions.

Figure 4 shows such an example for filtering L-R edges. The R-nodes that have strong connection with the L-node of interest are highlighted. As we expect, these R-nodes are nearby the L-node in the graph view since our force-directed layout algorithm assigns larger attractive forces to node pairs with higher edge weights.

5.4 Path Comparison and Region Comparison

Due to the occlusion-free 2D display of the FlowGraph, we enable the user to compare streamline clusters in terms of their paths going through different regions or compare spatial regions in terms of streamline clusters passing through them in a clear manner.

For path comparison, the user clicks a L-node in the graph and its corresponding paths passing through different R-nodes are highlighted. With hierarchical exploration, we allow comparing L-nodes at different levels of detail. Besides showing the actual paths the streamline cluster passing through, we also implement an algorithm similar to the *maximum spanning tree* algorithm to capture the main structure of the streamline cluster when the paths become cluttered. In addition, we filter out R-R edges of small weights as needed so that paths with very few streamlines passing through can be omitted. We draw undirected edges between R-nodes where the edge thickness indicates the strength of the path (i.e., the number of streamlines passing through in both directions). Multiple L-nodes can be selected simultaneously for path comparison. The paths are highlighted in the graph view and displayed in the spatial streamline view as well when the user mouses over the corresponding L-node. Furthermore, the user can also expend a L-node and check detail path information in a finer level. For this case, we do not perform path comparison. Instead, we display the detail path information using directed edges where we differentiate the flows coming in and going out of each region. For both directed and undirected path drawing, the user can always click one R-node along the path to highlight the R-node in both views. This allows the user to focus on particular regions of interest along the path.

For region comparison, the user clicks a R-node in the graph and the L-nodes passing through it are highlighted. Again, in conjunction with hierarchical exploration, we allow comparing R-nodes at different levels of detail. By selecting multiple R-nodes, the user can visually compare the streamline clusters passing through them in both the graph view and the streamline view.

Figure 5 (a) and (b) show path comparison with the two swirls data set. We can see that the graph view is highly correlated with the streamline view: the two swirls are well separated in the spatial domain and the corresponding L-nodes and R-nodes form two distinct connected components. Furthermore, highly related L-nodes and R-nodes are close to one another in the graph view. For example, the red streamline cluster is far away from the blue and purple clusters in the streamline view while the blue and purple clusters are neighbors. These relationships are well reflected in the graph view as well for intuitive exploration. Another example of path comparison with the solar plume data set is shown in Figure 5 (c) and (d). Unlike the streamline clusters in the two swirls data set, the two streamline clusters in the solar plume data set stretch a wide spatial range and their paths passing over many R-nodes. Six R-nodes shared in common by the two streamline clusters are highlighted in both views. The shared paths are blended of red and blue colors.

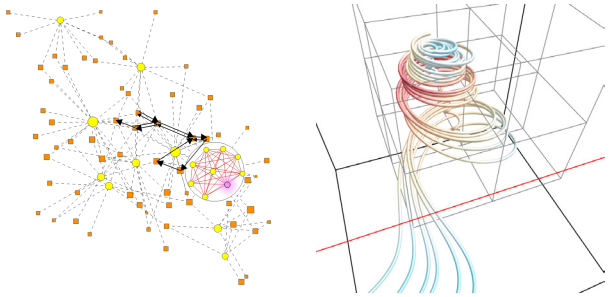


Figure 6: The detail path of a child L-node (shown in purple) of the tornado data set and the corresponding streamline cluster.

5.5 Graph Transition and Path Illustration

We introduce two different animation schemes to facilitate the understanding of the FlowGraph. The first scheme is *graph transition* where we show an animated transition from the compound graph to a single subgraph and vice versa. The motivation is to allow observation of the streamline cluster or spatial region subgraph in a less cluttered view. In addition, compared with the compound graph, the single subgraph layout for L-nodes (R-nodes) forms a better organization of node positions for observation of L-L edges (R-R edges). We start from the compound graph layout as the initial layout with either all L-nodes or R-nodes removed (the corresponding edges are removed as well). We then apply the force-directed graph layout algorithm to the remaining subgraph and generate a new layout for the subgraph. Animated transition is realized through linear interpolation of node positions over the duration of animation. The inverse from a subgraph to the compound graph is also provided.

The second scheme is *path illustration* where we show the detail path information for one streamline or a streamline cluster. For instance, Figure 6 shows an example of detail path. The directed black edges in the compound graph indicate the detail path information of the streamline cluster selected. The user can play an animation which indicates how the flow goes through the paths. If there are more than one path, we choose some prominent (i.e., the

data set	dimension	init. # lines	avg. # pts. per line	minimum spatial region	entropy threshold	GPU	CPU			graph
						entropy field	L-nodes	R-nodes	all edges	storage
car flow	$368 \times 234 \times 600$	600	185	$11 \times 7 \times 18$	0.2	0.109s	270.614s	0.010s	51.193s	25.5MB
computer room	$417 \times 345 \times 60$	800	173	$13 \times 10 \times 1$	0.9	0.136s	323.724s	0.035s	51.526s	36.2MB
five critical pts	$51 \times 51 \times 51$	500	112	$1 \times 1 \times 1$	1.0	0.069s	243.526s	0.020s	51.943s	37.1MB
hurricane	$500 \times 500 \times 100$	600	341	$15 \times 15 \times 3$	0.8	0.257s	230.816s	0.012s	51.435s	27.2MB
solar plume	$126 \times 126 \times 512$	600	100	$3 \times 3 \times 16$	1.1	0.130s	883.516s	0.030s	53.193s	30.1MB
supernova	$100 \times 100 \times 100$	500	184	$3 \times 3 \times 3$	0.8	0.079s	243.536s	0.020s	51.943s	23.8MB
tornado	$64 \times 64 \times 64$	500	295	$2 \times 2 \times 2$	1.0	0.070s	778.520s	0.029s	53.980s	23.8MB
two swirls	$64 \times 64 \times 64$	500	157	$2 \times 2 \times 2$	1.3	0.070s	324.975s	0.008s	50.986s	23.5MB

Table 1: The eight flow data sets experimented and their timing results for the FlowGraph construction. The entropy threshold is the entropy value of a spatial region divided by the number of voxels within that region.

longest) paths and play their animations simultaneously. The animation can be played in both the compound graph and a single subgraph. For the single streamline path animation, we also provide the function to traverse a streamline using animation in the streamline view. This streamline visualization is synchronized with the corresponding path animation shown in the graph view. Such an animation is very intuitive for the user to acquire a solid understanding of the relationships between the streamline or streamline clusters and the corresponding flow regions.

6 Results

We experimented our approach with eight flow data sets which are listed in Table 1. The car flow data set is from a simulation of the air flow around a car. The computer room data set is from a simulation of air flows inside a computer room. The five critical points data set [26] is a synthesized flow field consisting of two spirals, two saddles and one source. The hurricane data set is from a simulation of Hurricane Isabel, a strong hurricane in the west Atlantic region in September 2003. The solar plume data set is from a simulation of down-flowing solar plumes for studying the heat, momentum and magnetic field of the sun. The supernova data set is from a simulation of the explosion of stars. The tornado data set is from a simulation of a tornado event. Finally, the two swirls data set is from a simulation of swirls resulting from wake vortices.

We used a hybrid CPU-GPU solution in our computation with the following hardware configuration: Intel Core i7 quad-core CPU running at 3.20GHz, 24GB main memory and an nVidia GeForce GTX 580 graphics card. The parameter setting and timing performance are reported in Table 1. For all data sets, we randomly placed the seeds to trace streamlines over the field. The entropy calculation was performed in the GPU, while the FlowGraph construction was performed in the CPU. As we can see, the bottleneck step of the construction is to create the streamline hierarchy. The FlowGraph construction took up to 15 minutes to complete and the required storage for graph was quite affordable (less than 40MB). At runtime, all tasks including graph drawing, layout adjustment and user interaction in both views are interactive.

Selected FlowGraph results with individual exploration tasks have been shown in Figures 2 to 6. In the following, we present three case studies on three other data sets to demonstrate the capability of FlowGraph in assisting flow field exploration, path comparison and feature identification. To intuitively understand how the FlowGraph works and best evaluate its effectiveness, we refer readers to the accompanying video which shows the dual interaction on both the graph view and the streamline view.

Case Study 1 — Five Critical Points Data Set. For the five critical points data set, we experience how we can use the FlowGraph to easily identify these critical points from randomly traced streamlines that densely cover the field. In the first row of Figure 7, we show our exploration results that highlight three spatial regions that contain critical points. These spatial regions are important R-nodes in terms of centrality in the graph view. Normally, these R-nodes are close to the center of the graph and have strong connections to other nodes. As we can see in the streamline visualization, these three regions correspond to a spiral, a saddle and a source from left to right, respectively. In the second row of Figure 7, we select a R-node that has strong connection with its neighbor. Its corresponding spatial region is close to the center of the volume. The streamlines passing through this R-node are displayed. Since the number of streamlines displayed is fairly large, we further explore the child nodes of this R-node. Two child R-nodes and the streamlines passing through each of them are shown. It is clear that with the level-of-detail exploration, it becomes convenient for the user to explore the relationships between streamlines and spatial regions in an adaptive manner. This capability is very necessary in order to achieve flexible control when exploring large and complex 3D flow fields where dense streamlines are commonly exhibited throughout the entire volume.

Case Study 2 — Supernova Data Set. For the supernova data set, we first compare the paths of two streamline clusters. As shown in Figure 8 (a) and (b), these two streamline clusters both start from the volume boundary and get more intertwined as they get closer to the center. The compound graph view clearly shows the two R-nodes these two streamline clusters share in common. The highlighted path results also match the spatial arrangement of these two clusters. The paths start from the surrounding of the graph and advance to the center where the two clusters meet at the two spatial regions highlighted. In Figure 8 (c) and (d), we switch to the spatial region subgraph and show the path information of a single streamline. A R-node is further expanded to show the path information in the next level of detail. The corresponding spatial regions are highlighted in cyan. Observe how close the path drawn in the 2D graph view "matches" the 3D streamline view. In general, we find that drawing the subgraph which only consists of R-nodes and R-R edges forms a better arrangement of node positions. This helps the user build the connection between 2D paths and 3D streamlines between the views.

Case Study 3 — Car Flow Data Set. For the car flow data set, our goal is to identify spatial regions and streamline clusters that capture the essential interesting flow pattern passing through the car. In Figure 9, we can see that the FlowGraph exhibits an interesting layout: many L-nodes and R-nodes are pushed to the boundary of the drawing region. This is due to the fact that many of the streamlines we trace over the volume only form the straight pattern, i.e., they are simply passing by rather than passing through the car. These streamlines and spatial regions surround the interesting flow regions located around the center of the volume. These L-nodes and R-nodes only have a few connections to their neighboring nodes. In contrast, L-nodes and R-nodes around the center of the graph correspond to streamline clusters and spatial regions in the center of the volume. They have more connections to their neighboring nodes and are important nodes for our visual exploration. In Figure 9 (a), we select four R-nodes of interest. Eight L-nodes that have strong connection to the selected R-nodes are highlighted. The stream-

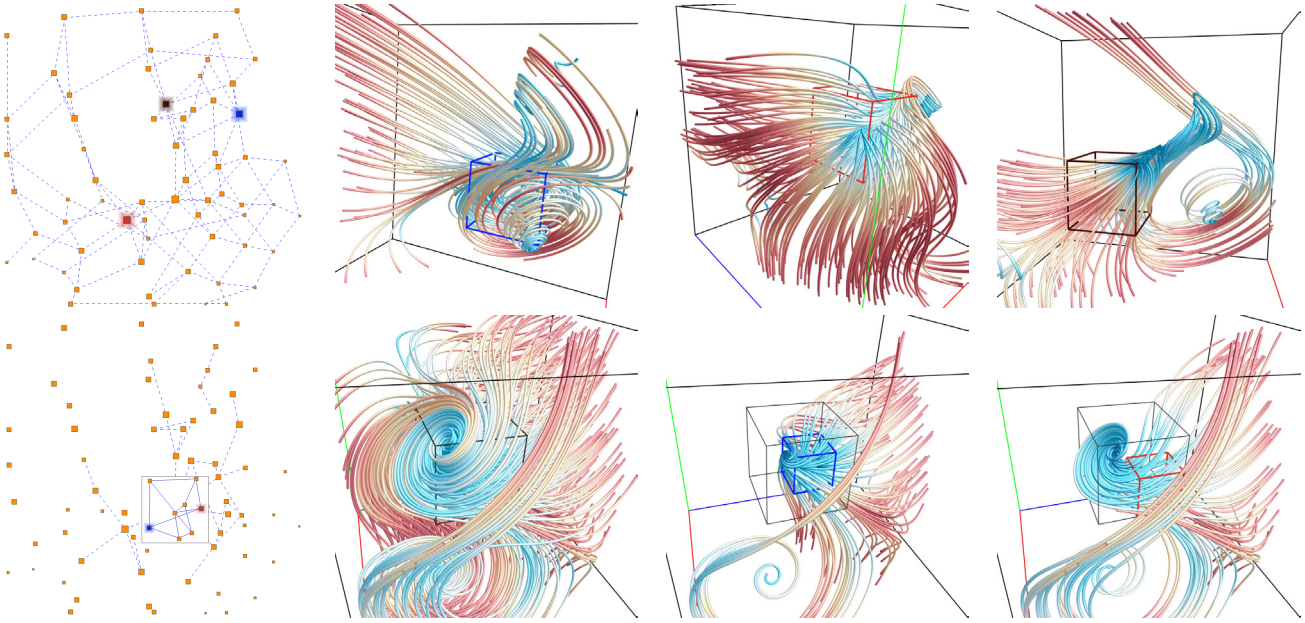
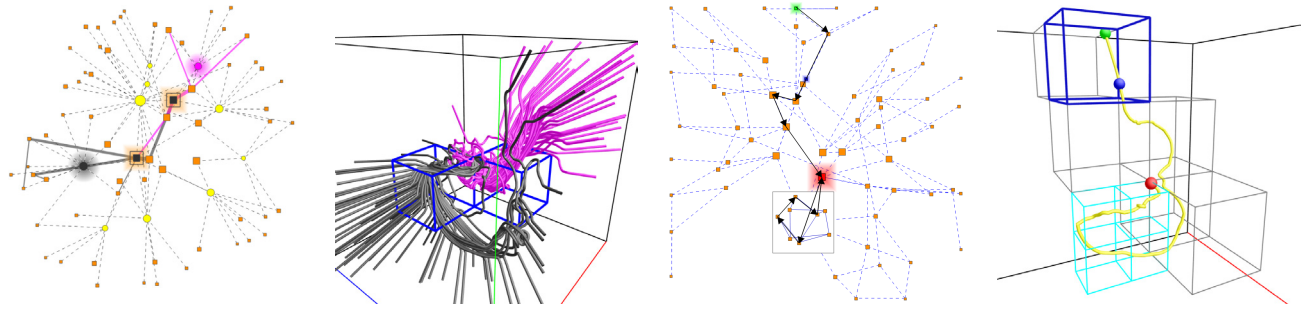


Figure 7: Exploring the five critical points data set. First row: three R-nodes are selected (shown in blue, red and brown) which correspond to the spatial regions each containing one critical point. Second row: filtering R-nodes based on the R-R edge weight identifies an important R-node. The streamlines passing through the parent R-node (shown in black) and two child R-nodes (shown in blue and red) are displayed.



(a) (b) (c) (d)

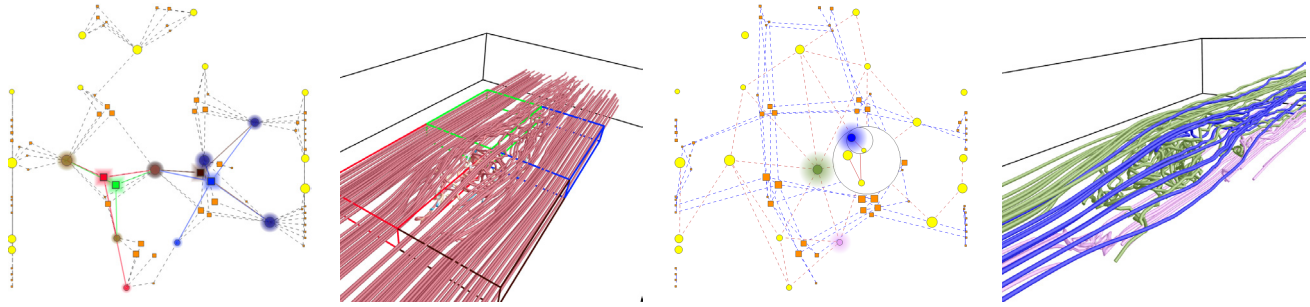
Figure 8: Exploration of the supernova data set using the FlowGraph. (a) and (b) show the path comparison of two streamline clusters (shown in black and magenta) in both views. Their shared spatial regions are also highlighted. (c) and (d) show the snapshot of path animation of a single streamline over spatial regions with different levels of detail. Green, red and blue squares (graph view) and spheres (streamline view) indicate the starting, ending and current animation points, respectively.

line view shown in (b) clearly indicates the correspondence of these nodes to interesting flow regions. In (c) and (d), we further explore three L-nodes and filter out streamline clusters at two different levels of detail that well capture the flow pattern passing through the car. With the visual guidance of the FlowGraph and dual interaction with the streamline view, exploring the underlying flow field to identify features of interest becomes more intuitive, convenient and effective.

7 Conclusion and Future Work

We have presented the FlowGraph, a new graph-based visual representation that explicitly encodes the streamline clusters, spatial regions and their hierarchical relationships to assist flow field exploration and interrogation. The main motivation to generate such a representation is to address the intrinsic difficulty when visualizing and understanding 3D streamlines. As we know, 3D streamlines normally create dense distribution over the space, which is especially true for large and complex 3D flow fields. By transforming the streamlines, spatial regions and their interconnection to a 2D space, we allow occlusion-free observation, navigation and interaction with the graph view and make connection to the streamline view for effective visual exploration. Our work falls into the category of visual analytics for scientific visualization: extracting essential information or relationships from scientific data sets to enable analytical reasoning facilitated by interactive visual interfaces. Even though the FlowGraph is an abstract representation of the underlying flow field, our results show strong evidence with multiple data sets that it is easy to understand the graph and perform the tasks accordingly through visual encoding such as node size, visual hints such as node centrality, and interactive filtering such as edge pruning. Our experience shows that through brushing and linking, the user can quickly build the connections between the views. Once such connections are built, the user will have a good understanding on how to work with both views effectively to achieve different visual exploration goals.

We would like to extend our FlowGraph to time-varying flow field visualization and exploration. For time-varying flow fields, different kinds of field lines can be produced such as pathlines, streaklines and timelines. Pathlines are the obvious choice to build the FlowGraph for time-varying flow fields, while the introduction of the time dimension may require us to develop a time-varying



(a) (b) (c) (d)

Figure 9: Exploring the interesting flow pattern in the car flow data set. (a) and (b) show four important R-nodes (shown in red, green, blue and brown) and eight L-nodes that have strong connection to the R-nodes of interest. From these eight L-nodes, (c) and (d) show further selection of three L-nodes (one at the next level of the hierarchy) to capture the main flow structure passing through the car.

graph for better capturing time-dependent flow features. This poses new challenges in hierarchy construction and graph visualization, as well as graph understanding and visual interpretation.

Acknowledgements

This work was supported in part by the U.S. National Science Foundation through grants IIS-1017935, DUE-1105047 and CNS-1229297. We thank the anonymous reviewers for their insightful comments.

References

[1] A. Brambilla, R. Carnecky, R. Peikert, I. Viola, and H. Hauser. Illustrative flow visualization: State of the art, trends and challenges. In *Eurographics State-of-the-Art Reports*, pages 75–94, 2012.

[2] C.-M. Chen, L. Xu, T.-Y. Lee, and H.-W. Shen. A flow-guided file layout for out-of-core streamline computation. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 145–152, 2012.

[3] C. D. Correa, D. Silver, and M. Chen. Illustrative deformation for data exploration. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1320–1327, 2007.

[4] W. Cui, Y. Wu, S. Liu, F. Wei, M. X. Zhou, and H. Qu. Context preserving dynamic word cloud visualization. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 121–128, 2010.

[5] H. Doleisch, M. Gasser, and H. Hauser. Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of Eurographics/IEEE TCVG Symposium on Visualization*, pages 239–248, 2003.

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

[7] A. L. Fuhrmann and M. E. Gröller. Real-time techniques for 3D flow visualization. In *Proceedings of IEEE Visualization Conference*, pages 305–312, 1998.

[8] R. Gasteiger, M. Neugebauer, O. Beuing, and B. Preim. The FLOWLENS: A focus-and-context visualization approach for exploration of blood flow in cerebral aneurysms. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2183–2192, 2011.

[9] B. Heckel, G. H. Weber, B. Hamann, and K. I. Joy. Construction of vector field hierarchies. In *Proceedings of IEEE Visualization Conference*, pages 19–25, 1999.

[10] E. Heiberg, T. Ebbers, L. Wigström, and M. Karlsson. Three-dimensional flow characterization using vector pattern matching. *IEEE Transactions on Visualization and Computer Graphics*, 9(3):313–319, 2003.

[11] R. S. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser, and H. Hagen. Visual analysis and exploration of fluid flow in a cooling jacket. In *Proceedings of IEEE Visualization Conference*, pages 623–630, 2005.

[12] R. S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F. H. Post, and D. Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–222, 2004.

[13] O. Mattausch, T. Theußl, H. Hauser, and M. E. Gröller. Strategies for interactive exploration of 3D flow using evenly-spaced illuminated streamlines. In *Proceedings of Spring Conference on Computer graphics*, pages 213–222, 2003.

[14] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen. Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum*, 29(6):1807–1829, 2010.

[15] B. Moberts, A. Vilanova, and J. J. van Wijk. Evaluation of fiber clustering methods for diffusion tensor imaging. In *Proceedings of IEEE Visualization Conference*, pages 65–72, 2005.

[16] B. Nouanesengsy, T.-Y. Lee, and H.-W. Shen. Load-balanced parallel streamline generation on large scale vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):1785–1794, 2011.

[17] Z. Peng and R. S. Laramee. Higher dimensional vector field visualization: A survey. *Theory and Practice of Computer Graphics*, pages 149–163, 2009.

[18] C. Rössl and H. Theisel. Streamline embedding for 3D vector field exploration. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):407–420, 2012.

[19] T. Salzbrunn, H. Jänicke, T. Wischgoll, and G. Scheuermann. The state of the art in flow visualization: Partition-based techniques. In *Proceedings of Simulation and Visualization Conference*, pages 75–92, 2008.

[20] M. Schlemmer, M. Heringer, F. Morr, I. Hotz, M.-H. Bertram, C. Garth, W. Kollmann, B. Hamann, and H. Hagen. Moment invariants for the analysis of 2D flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1743–1750, 2007.

[21] D. Schroeder, D. Coffey, and D. F. Keefe. Drawing with the flow: A sketch-based interface for illustrative visualization of 2D vector fields. In *Proceedings of ACM SIGGRAPH/Eurographics Sketch-Based Interfaces and Modeling*, pages 49–56, 2010.

[22] J. R. Shewchuk. Triangle: Engineering a 2D quality mesh generator and Delaunay triangulator. In *Proceedings of ACM Workshop on Applied Computational Geometry*, pages 203–222, 1996.

[23] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proceedings of IEEE Visualization Conference*, pages 35–42, 1999.

[24] J. Wei, C. Wang, H. Yu, and K.-L. Ma. A sketch-based interface for classifying and visualizing vector fields. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 129–136, 2010.

[25] L. Xu and H.-W. Shen. Flow web: A graph based user interface for 3D flow field exploration. In *SPIE Proceedings of Visualization and Data Analysis*, 2010.

[26] X. Ye, D. Kao, and A. Pang. Strategy for seeding 3D streamlines. In *Proceedings of IEEE Visualization Conference*, pages 471–478, 2005.

[27] H. Yu, C. Wang, C.-K. Shene, and J. H. Chen. Hierarchical streamline bundles. *IEEE Transactions on Visualization and Computer Graphics*, 18(8):1353–1367, 2012.