

A Graph-Based Interface for Visual Analytics of 3D Streamlines and Pathlines

Jun Ma, *Student Member, IEEE*, Chaoli Wang, *Member, IEEE*,
Ching-Kuang Shene, *Member, IEEE Computer Society*, and Jingfeng Jiang

Abstract—Visual exploration of large and complex 3D steady and unsteady flow fields is critically important in many areas of science and engineering. In this paper, we introduce FlowGraph, a novel compound graph representation that organizes field line clusters and spatiotemporal regions hierarchically for occlusion-free and controllable visual exploration. It works with any seeding strategy 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 field line clusters, spatiotemporal regions and their interconnection in the transformed space. FlowGraph not only provides a visual mapping that abstracts field line clusters and spatiotemporal 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 standard field line view, we demonstrate the effectiveness of FlowGraph with several visual exploration and comparison tasks that cannot be well accomplished using the field line view alone. We also perform an empirical expert evaluation to confirm the usefulness of this graph-based technique.

Index Terms—Flow visualization, streamlines, pathlines, graph representation, visual interface, visual analytics.

1 INTRODUCTION

For more than two decades, flow visualization has been a central topic in scientific visualization and a variety of techniques, including glyph-based [20], texture-based [14], integration-based [17], partition-based [22], illustration-based [2], and surface-based [7] 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 [2], [17]. Furthermore, even though streamlines can be organized into a hierarchy to

facilitate the understanding [11], [26], [30], 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. Pathlines are even more challenging than streamlines due to the addition of the time dimension. In this case, we need to examine and explore pathlines and their spatiotemporal relationships.

In this extended version of our IEEE PacificVis 2013 paper [15], we present FlowGraph, a visual representation and an interface for effective exploration and analytics of a 3D flow field. The design target of FlowGraph is to address the intrinsic limitations of 3D occlusion and lack of control when using the standalone field line view for field line exploration, comparison and examination. In particular, we seek deeper meanings and richer values of leveraging such an abstract graph in flow field analysis. In our view, this greater extent of integrating information visualization techniques for scientific visualization represents a significant advance in visual analytics of flow field data. Our solution works with both streamlines for steady flow fields and pathlines for unsteady flow fields. In conjunction with the standard view of field lines, FlowGraph transforms field line clusters and spatiotemporal regions into a compound hierarchical graph representation to support effective relationship overview and detailed exploration.

We specifically design a set of functions that enable hierarchical exploration of field line clusters, spatiotemporal regions and their interconnection, detailed comparison among field line clusters in terms of their paths passing through different spatiotemporal regions, and close examination of spatiotemporal regions by comparing different field line clusters passing through them. Through brushing and linking, the user can easily make connection between the graph view and the field line view. Animation is used to help intuitive comprehension of graph transition and path illustration. A graph layout algorithm

- J. Ma, C. Wang, and C.-K. Shene are with the Department of Computer Science, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. E-mail: {jumm, chaoliw, shene}@mtu.edu.
- J. Jiang is with the Department of Biomedical Engineering, Michigan Technological University, 1400 Townsend Drive, Houghton, MI 49931. E-mail: jjiang1@mtu.edu.

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 field line cluster or spatiotemporal region subgraph, allowing observation of the subgraphs in a less cluttered view.

From Sections 3 to 5, we present FlowGraph definition and construction, drawing, exploration and interrogation in the context of streamlines. In Section 6, we extend FlowGraph to handle pathlines, focusing on the differences with respect to streamlines and explaining our strategy and solution. To demonstrate the effectiveness of FlowGraph, we perform several case studies on flow field data sets of various characteristics (Section 7) and conduct an empirical expert evaluation (Section 8). Our results show that 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. Extension of FlowGraph to handle pathlines and feedback from a domain expert are new to the journal version of this work, and thus significantly extends the work in the IEEE PacificVis 2013 paper [15].

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 presented. For instance, Heiberg et al. [12] located, identified and visualized a set of predefined structures in 3D flows using vector pattern matching. Schlemmer et al. [23] 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 [21] 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. [24] 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. [27] 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 created streamline templates hierarchically to support on-the-fly partial streamline matching in a progressive manner.

Focus+Context Flow Visualization. Researchers also explored different focus+context techniques to enable greater control in visual examination of flow fields. Fuhrmann and Gröller [9] 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 [16]. Laramée et al. [13] leveraged feature-based techniques [6] 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

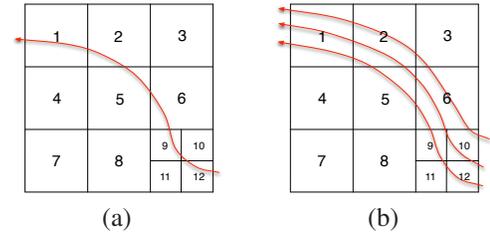


Fig. 1. Illustration of L-node signature with a 2D space partitioning example. (a) the signature of the streamline is an ordered sequence (12, 10, 9, 6, 5, 2, 1). (b) the signature of the streamline cluster is an unordered set (1, 2, 3, 5, 6, 9, 10, 11, 12).

interactive thresholding. Correa et al. [4] 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. [10] designed 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 [28] 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 [3], [19]. 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 FlowGraph (without L-nodes, L-L edges and L-R edges). FlowGraph allows the user to fully explore their relationships through interacting with the graph view and making connection to the streamline view. Furthermore, our FlowGraph also works well for pathlines traced from unsteady flow fields, which was not shown in the work of flow web.

3 FLOWGRAPH DEFINITION AND CONSTRUCTION FOR STEADY FLOW FIELD

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

- *R-nodes*: An R-node represents a spatial region. We partition the volume space hierarchically using an 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 an 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*: An 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*: An 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*: An L-R edge is formed between an L-node and an 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 an 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. Refer to Table 1 for the parameter values we set for each data set. This produces a spatial partition similar to an adaptive mesh refinement (AMR) grid [1].

3.2 Streamline Similarity

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 an approximation of the *mean of the closest point distance* (MCP) [18] 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 incurs a much lower computation cost than the MCP does. Furthermore, since the MCR is always computed by using regions at the finest level, its accuracy is also acceptable as judged from the generated streamline clustering results. The final similarity between two streamlines l_1 and l_2 is defined as

$$\Phi(l_1, l_2) = \frac{\text{LCS}(l_1, l_2)}{\max(|l_1|, |l_2|)} - \frac{\text{MCR}(l_1, l_2)}{\text{MCR}_{\max l}}, \quad (1)$$

where $\max(|l_1|, |l_2|)$ is the maximum signature size of l_1 and l_2 , and $\text{MCR}_{\max l}$ 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 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

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

where $\max(|c_1|, |c_2|)$ is the maximum signature size of c_1 and c_2 , and $\text{MCR}_{\max c}$ 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.

3.3 Streamline Hierarchy Construction

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

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

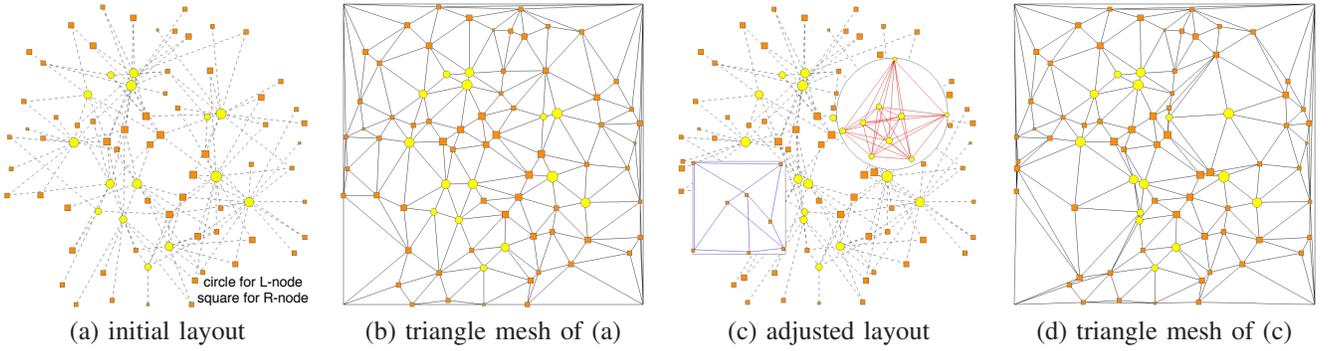


Fig. 2. (a) the initial layout is produced using the force-directed graph layout algorithm. The size of each node in the graph is proportional to the number of children within. (b) the triangle mesh produced from the initial node positions. (c) the adjusted layout after two nodes are selected and expanded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

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 v_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 (the number is usually 1/10 to 1/5 of the number of L-nodes in the lower 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 clustering at the next level. We repeat the entire process until a certain number of streamline levels is created.

In practice, for constructing FlowGraph, it is desirable for spatial regions or streamline clusters to have three to five levels in their respective hierarchy. This is suggested through empirical observations of the resulting graph's size and 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 FlowGraph in an efficient way and maintain a good balance between clarity and complexity.

4 FLOWGRAPH DRAWING FOR STEADY FLOW FIELD

We apply the Fruchterman-Reingold algorithm, a classical force-directed graph layout algorithm [8] 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 cross *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 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 FlowGraph. To achieve stable update, we apply a triangulation scheme [25] 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 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 expanded 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 [5], we consider four kinds of forces to reposition the nodes to reduce their overlap while maintaining the topology of the coarsest level of 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 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. 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

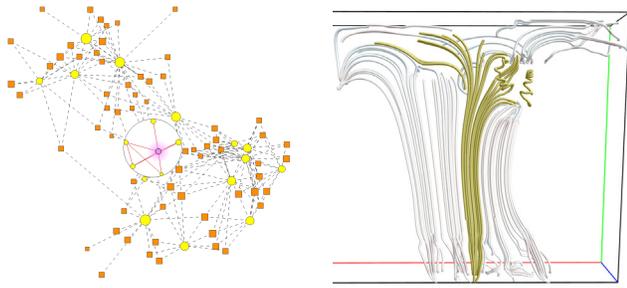


Fig. 3. An L-node is expanded in the computer room data set and one of its child nodes is shown in purple. The corresponding child and parent streamline clusters are shown in gold and white, respectively.

while the global structure of FlowGraph is still preserved. We apply this same layout adjustment strategy recursively to nodes at different hierarchical levels.

5 FLOWGRAPH EXPLORATION AND INTERROGATION FOR STEADY FLOW FIELD

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. For instance, if an R-node has connections 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 an 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

With FlowGraph, the user can select a node of interest, expand it to see its next level of detail recursively, and make a connection to the spatial streamline view. 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 layout of FlowGraph 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

We dynamically connect the graph view and the streamline view together through brushing and linking: when the user

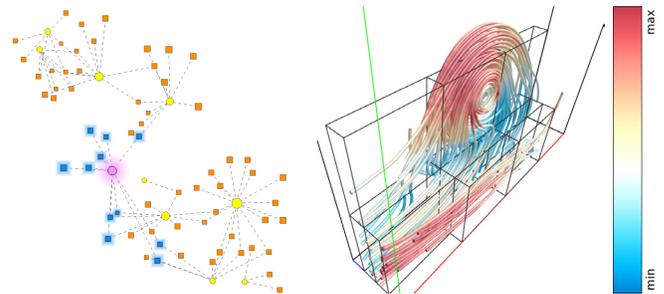


Fig. 4. Filtering L-R edges by weight in FlowGraph highlights eleven R-nodes (shown in blue) that have strong connection with the L-node of interest (shown in purple) in the hurricane data set. Velocity magnitudes are mapped to streamline colors.

clicks an L-node (R-node) in the graph view, its corresponding streamline cluster (spatial region) is highlighted in the streamline view, and vice versa. 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 meanwhile, the corresponding paths passing R-nodes will also be highlighted in the graph view. Similar hints on the corresponding streamline clusters will be provided when a spatial region is selected. Through brushing and linking, especially combined with hierarchical exploration, the user 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 detailed 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 FlowGraph. 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 FlowGraph, the user can 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 an 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

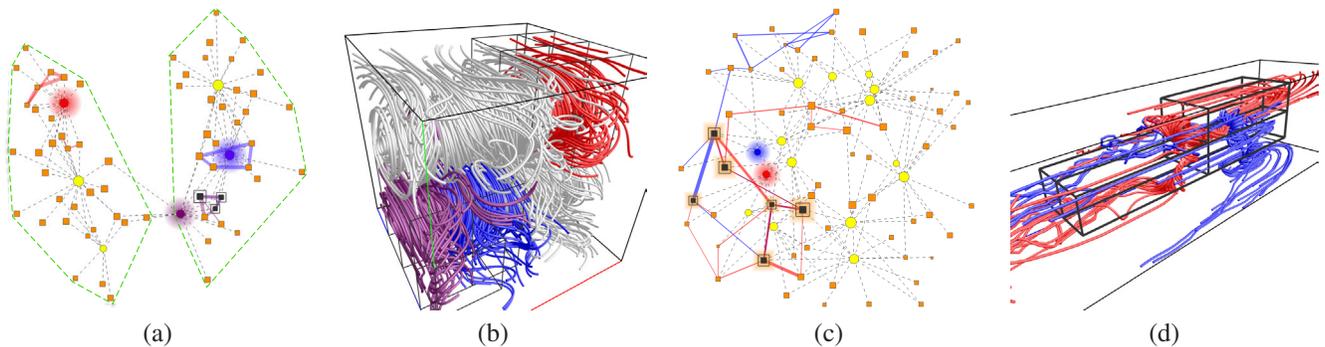


Fig. 5. (a) and (b) show path comparison for three streamline clusters shown in red, blue and purple, respectively of the two swirls data set. Observe that the two swirls are well separated in the graph view as indicated by the green dashed lines. Three R-nodes shared by the blue and purple L-nodes are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

filter out R-R edges of small weights to omit paths with very few streamlines passing through. 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.

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.

For region comparison, the user clicks an 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 views.

5.5 Graph Transition and Path Illustration

We introduce two different animation schemes to facilitate the understanding of 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 observing L-L edges (R-R edges).

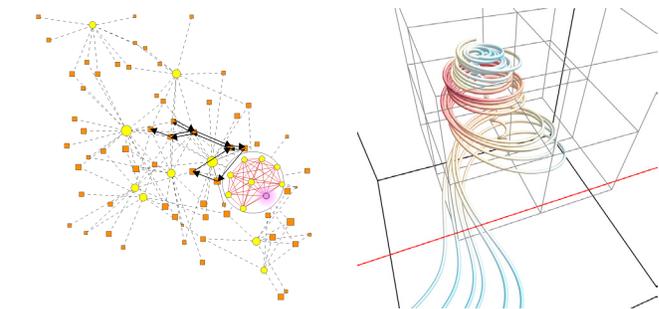


Fig. 6. The detail path of a child L-node (shown in purple) of the tornado data set and the corresponding streamline cluster.

The second scheme is *path illustration* where we show the detail path information for one streamline or a streamline cluster. 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 traverses the paths. 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. This addition is intuitive for the user to acquire a solid understanding of the relationships between the streamline or streamline cluster and its corresponding flow regions.

6 FLOWGRAPH EXTENSION TO UNSTEADY FLOW FIELD

Due to the high-dimensional nature of unsteady flow fields, providing a visual exploration tool to explicitly show the relationships between pathline clusters and their corresponding spatiotemporal regions becomes a major challenge. To overcome this problem, in this section we extend our FlowGraph to handle 3D unsteady flow fields.

6.1 FlowGraph Definition and Construction

Our FlowGraph for unsteady flow fields is also a compound hierarchical graph that consists of two kinds of nodes (R-node

and L-nodes) and three kinds of edges (R-R edges, L-L edges and L-R edges). We modify the definitions for these nodes and edges as follows. An R-node now represents a spatiotemporal region. We use a 4D octree (i.e., 16-tree) to partition the unsteady flow data from both spatial and temporal dimensions simultaneously. Each leaf R-node maintains a list recording all pathlines going through the corresponding spatial region within a particular time interval. By treating 3D pathlines as 4D streamlines, we construct L-nodes in the same way as we do for 3D streamlines. Specifically, a leaf L-node represents a single pathline, and a non-leaf L-node indicates a pathline cluster. Furthermore, each L-node records the range of time interval for the pathline or the pathline cluster it corresponds to. Each L-node also keeps a signature it traverses through where the spatial regions are replaced by the spatiotemporal regions. In terms of edges, by replacing streamlines and spatial regions with pathlines and spatiotemporal regions, we follow the same definitions in Section 3 to define R-R edges, L-L edges and L-R edges.

6.2 Space-Time Hierarchy Construction

Similar to the octree partition (Section 3.1), we obtain the space-time hierarchy by partitioning the spatiotemporal domain evenly in a top-down manner using 16-tree. Specifically, we treat the unsteady flow data as a 4D continuous space which contains x , y , z and t (time) components. Starting from the entire 4D data set, we evenly divide it along each dimension at each iteration. The partition termination criteria is still based on the entropy value of the spatiotemporal region or the given threshold for the smallest spatiotemporal size. Intuitively, each partitioned region is a spatiotemporal region group which occupies a cubic volume in space and spans across a certain time interval. In practice, over hundreds of thousands of leaf regions could be generated. We therefore use a 4D tree data structure to store all the leaf regions for fast access.

6.3 Pathline Hierarchy Construction

Since we treat 3D pathlines as 4D streamlines, pathline hierarchy construction follows the same scheme in Sections 3.2 and 3.3. The difference is that similarity computation is now based on both spatial and temporal information of the corresponding pathlines or pathline clusters. We compute the LCS between two pathline L-nodes' signatures as usual. In terms of MCR computation, rather than only considering spatial distance between two regions, we compute the distance based on both spatial and temporal information by using each region's center as a 4D point (x, y, z, t) . We follow the same solutions to group pathlines level by level for hierarchy construction and to select the representative pathline from each cluster.

6.4 FlowGraph Drawing

Several new features are provided to highlight temporal information for FlowGraph drawing. Node and edge drawing follows the same style as before. Color saturation is used to distinguish nodes based on their time spans across the entire time sequence of the data set. Specifically, a node with an early (later) time span is drawn in low (high) saturation. In

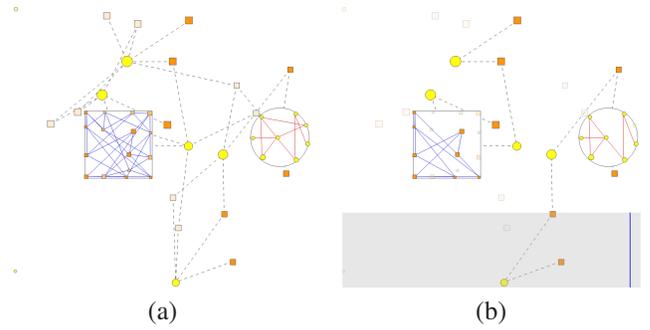


Fig. 7. (a) is FlowGraph for the unsteady solar plume data set. An L-node and an R-node are expanded. Nodes with low (high) saturated colors indicate early (later) temporal regions or pathlines. (b) shows the graph with the timeline bar where the blue line shows the current time step. Semitransparent nodes indicate the corresponding spatiotemporal regions or pathlines whose time spans do not cover the current time step.

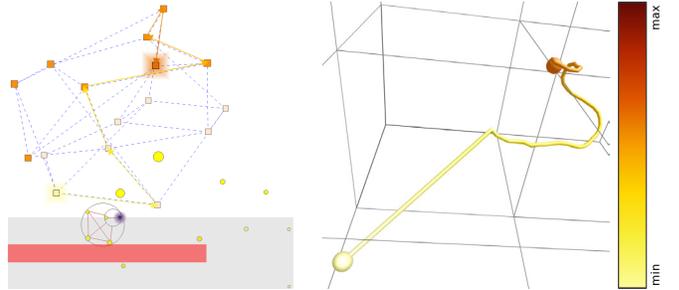


Fig. 8. The colored single path for the unsteady supernova data set. The red interval in the timeline bar indicates the time span of the selected pathline. The same colormap is used for both the pathline and its path in the graph view. Yellow indicates the earliest time step and brown indicates the latest time step.

terms of layout computation, we still apply the Fruchterman-Reingold algorithm where an R-R edge connects two neighboring regions based on both spatial and temporal information. Figure 7 (a) shows such an example. To help the user explore the graph at a specific time step, we provide a timeline bar to indicate the current time step and filter out graph nodes whose time spans do not cover it by making them semitransparent. An example is shown in Figure 7 (b) where the horizontal direction of the timeline bar from left to right corresponds to early and later time steps.

6.5 FlowGraph Exploration and Interrogation

Our FlowGraph for pathlines and spatiotemporal regions keeps all the exploration and interrogation functions for streamlines and spatial regions. Furthermore, by plugging the time information into the graph, FlowGraph conveys more information and provides the user with more flexibility to observe and explore the unsteady flow field. For example, since we use the entropy to determine the size of an R-node, two R-nodes which occupy the same spatial region but cover different time spans could indicate the change of entropy in the same spatial region

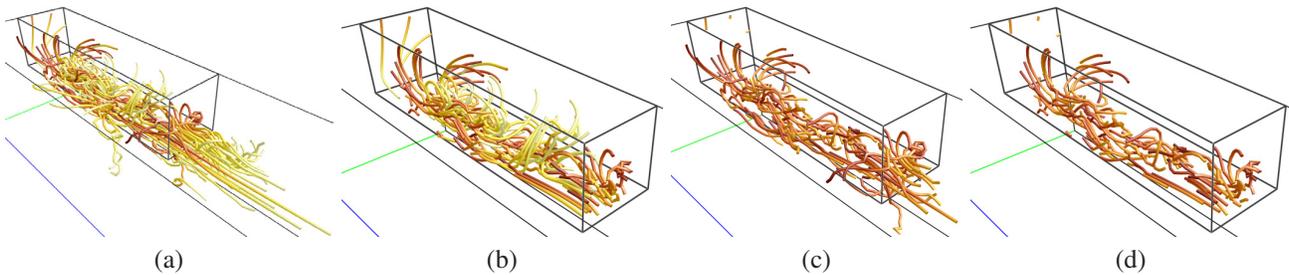


Fig. 9. (a) shows pathlines going through the selected spatiotemporal region for the unsteady solar plume data set. (b) to (d) show pathline segments inside of the region’s spatial boundary, temporal boundary, and spatial and temporal boundaries, respectively.

over time. For hierarchical exploration, when the user selects a node, we also show its time span in red in the timeline bar. We provide a time slider to help the user select a spatiotemporal region at any specific time step. For path comparison and region comparison, when demonstrating the path for a single pathline in the graph view, instead of drawing the path in black, we colorize the path using the same color mapping for pathline drawing to show the time correspondence. An example is shown in Figure 8. In order to differentiate pathlines from streamlines, we use a different colormap for pathline drawing.

Besides these existing functions for FlowGraph, we add the following new features to handle graph exploration involving the temporal aspect:

Pathline Spatial and Temporal Filtering. We provide pathline filtering option to allow the user to focus on the pathlines for a specific spatial region or time interval. Basically, when the user selects a region in the pathline view, FlowGraph shows all the pathlines passing through this region by default. However, sometimes it could be difficult for the user to observe clearly the flow patterns inside of the selected region. Possible reasons are that there may be too many pathlines going through the selected region and these pathlines may also pass through some other regions and thus make the view cluttered. To alleviate this issue, we render the portions of pathlines that are only inside of the region by filtering out pathlines segments that are outside of the region’s spatial or temporal boundary. Figure 9 shows an example of this filtering.

Pathlet Rendering and Animation. Rather than only showing the entire pathline indicating the whole trajectory of a particle, we also draw the pathlet to show a segment of the trajectory over a short time interval. The arrow of the pathlet indicates the current flow direction. Figure 10 (a) shows the pathlets for the unsteady solar plume data set. By utilizing pathlet rendering, our FlowGraph allows the user to only focus on the flow patterns in some specific time interval. Moreover, using a time slider, the user can obtain pathlet animation to indicate the evolution of flow over time. Please refer to the accompanying video for the animation. Pathlet rendering and animation could be combined with other functions to provide the user with a more comprehensive understanding of the flow fields. For example, when the user selects a region and wants to observe the corresponding pathlines going through this region, she can first applies pathline spatial and temporal filtering and then uses pathlets to demonstrate how the flow patterns change

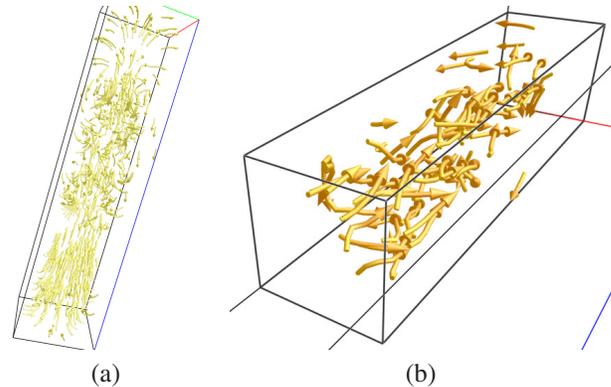


Fig. 10. (a) shows all the pathlets in a specific time interval for the unsteady solar plume data set. (b) shows the pathlets inside of a selected spatiotemporal region. The same colormap shown in Figure 8 is used to indicates the time steps of the pathlets.

inside of this region over its time span. Figure 10 (b) shows such an example.

7 RESULTS

We experimented our approach with eight steady flow data sets and three unsteady 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, courtesy of Alex Pang at the University of California, Santa Cruz, is a synthesized flow field consisting of two spirals, two saddles and one source [29]. The hurricane data set, courtesy of IEEE Visualization 2004 Contest, is from a simulation of Hurricane Isabel, a strong hurricane in the west Atlantic region in September 2003. The solar plume data set, courtesy of John Clyne at the National Center for Atmospheric Research, is from a simulation of down-flowing solar plumes for studying the heat, momentum and magnetic field of the sun. The supernova data set, courtesy of Anthony Mezzacappa at Oak Ridge National Laboratory and John Blondin at North Carolina State University, is from a simulation of the explosion of stars. The tornado data set, courtesy of Roger Crawfis at The Ohio State University, is procedurally generated from a piece of code. Finally, the two swirls data set is from a simulation of swirls resulting from wake vortices.

TABLE 1

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

data set	dimension	init. # lines	avg. # pts. per line	minimum region	entropy threshold	GPU	CPU			graph storage
						entropy field	L-nodes	R-nodes	all edges	
car flow	368 × 234 × 600	600	185	11 × 7 × 18	0.2	0.109s	270.614s	0.010s	51.193s	25.5MB
computer room	417 × 345 × 60	800	173	13 × 10 × 1	0.9	0.136s	323.724s	0.035s	51.526s	36.2MB
five critical pts	51 × 51 × 51	500	112	1 × 1 × 1	1.0	0.069s	243.526s	0.020s	51.943s	37.1MB
hurricane	500 × 500 × 100	600	341	15 × 15 × 3	0.8	0.257s	230.816s	0.012s	51.435s	27.2MB
solar plume	126 × 126 × 512	600	100	3 × 3 × 16	1.1	0.130s	883.516s	0.030s	53.193s	30.1MB
supernova	100 × 100 × 100	500	184	3 × 3 × 3	0.8	0.079s	243.536s	0.020s	51.943s	23.8MB
tornado	64 × 64 × 64	500	295	2 × 2 × 2	1.0	0.070s	778.520s	0.029s	53.980s	23.8MB
two swirls	64 × 64 × 64	500	157	2 × 2 × 2	1.3	0.070s	324.975s	0.008s	50.986s	23.5MB
hurricane	500 × 500 × 100 × 48	800	37	15 × 15 × 3 × 1	0.8	14.812s	1539.850s	91.694s	280.981s	175.1MB
solar plume	126 × 126 × 512 × 29	600	25	7 × 7 × 32 × 1	0.2	3.558s	379.973s	20.035s	31.425s	122.2MB
supernova	216 × 216 × 216 × 105	500	58	3 × 3 × 3 × 3	0.8	12.532s	1359.309s	85.760s	112.387s	102.5MB

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 steady data sets, we randomly placed the seeds to trace streamlines over the field. For unsteady data sets, seeds are randomly placed at the first time step for pathline tracing. The entropy calculation was performed in the GPU, while FlowGraph construction was performed in the CPU. As we can see, the bottleneck step of the construction is to create the streamline or pathline hierarchy. For steady flow fields, it took up to 15 minutes to construct FlowGraph and the required storage for graph was quite affordable (less than 40MB). For unsteady flow fields, since the number of spatiotemporal regions after the 16-tree partition is much larger than the number of spatial regions in steady flow fields, the time for R-node hierarchy computation increased dramatically and could reach around 1.5 minutes. The total graph construction time was up to 30 minutes and the storage space also increased to up to 175MB. 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 10. In the following, we present three case studies on three other steady flow data sets to demonstrate the capability of FlowGraph in assisting flow field exploration, path comparison and feature identification. We also give two case studies on two unsteady flow data sets to show the exploration of relationships between pathlines and spatiotemporal regions. To intuitively understand how 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 field line view.

Case Study 1 — Five Critical Points Data Set. For the five critical points data set, we experience how we can use FlowGraph to easily identify these critical points from randomly traced streamlines that densely cover the field. In the first row of Figure 11, 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 11, we select an 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 — Steady Supernova Data Set. For the supernova data set, we first compare the paths of two streamline clusters. As shown in Figure 12 (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 12 (c) and (d), we switch to the spatial region subgraph and show the path information of a single streamline. An 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 13, we can see that 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

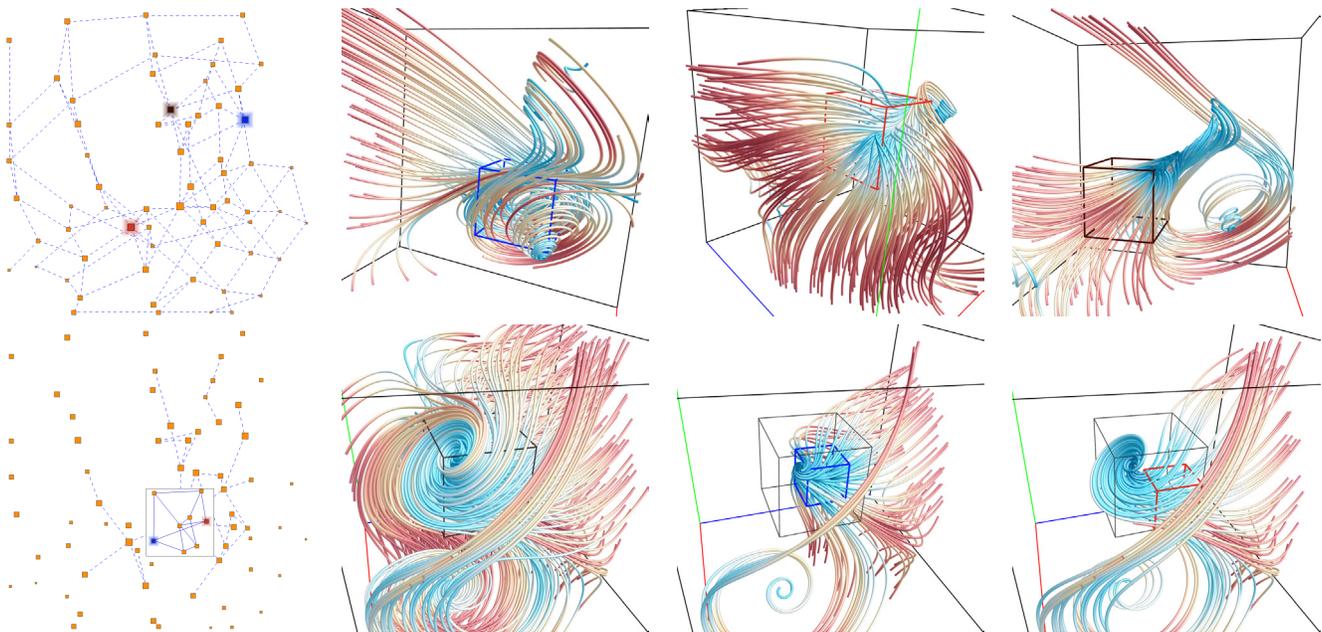


Fig. 11. Exploring the five critical points data set. First row: three R-nodes are selected (shown in blue, red and brown) which correspond to the spatial regions each containing one critical point. Second row: filtering R-nodes based on the R-R edge weight identifies an important R-node. The streamlines passing through the parent R-node (shown in black) and two child R-nodes (shown in blue and red) are displayed.

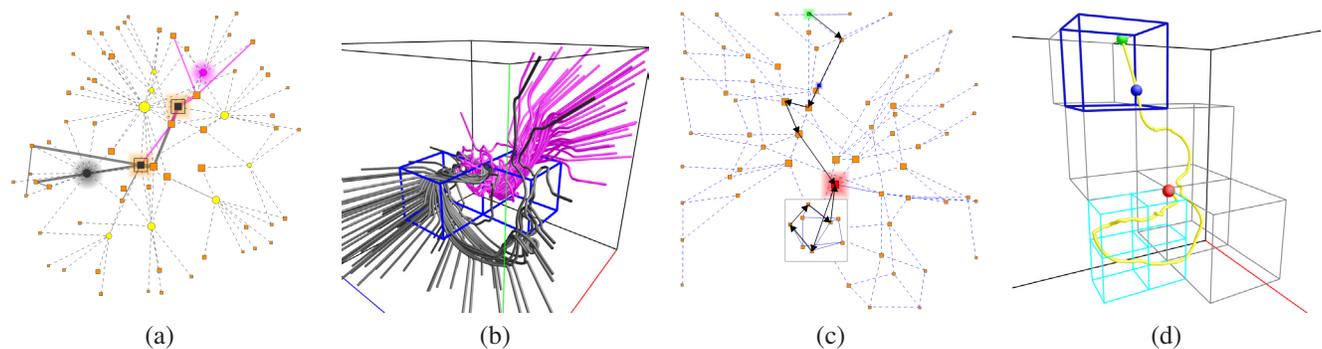


Fig. 12. Exploration of the steady supernova data set using FlowGraph. (a) and (b) show the path comparison of two streamline clusters (shown in black and magenta) in both views. Their shared spatial regions are also highlighted. (c) and (d) show the snapshot of path animation of a single streamline 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.

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 13 (a), we select four R-nodes of interest. Eight L-nodes that have strong connections to the selected R-nodes are highlighted. The streamline 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 FlowGraph and dual interaction with the streamline view, exploring the underlying flow field to identify features of interest becomes more intuitive, convenient and effective.

Case Study 4 — Unsteady Supernova Data Set. For the unsteady supernova data set, we first utilize the relationships between R-nodes and L-nodes combined with pathlet animation to detect one sink at the core of the supernova. In Figure 14 (a) and (c), we highlight two R-nodes which occupy the same spatial region but cover different time spans. The L-nodes connecting to them are also shown in the same color. Two L-nodes are expanded to provide pathline cluster observation at finer levels of detail. We can see that the blue R-node from the early time span has more connections to the L-nodes than the red R-node from the later time span.

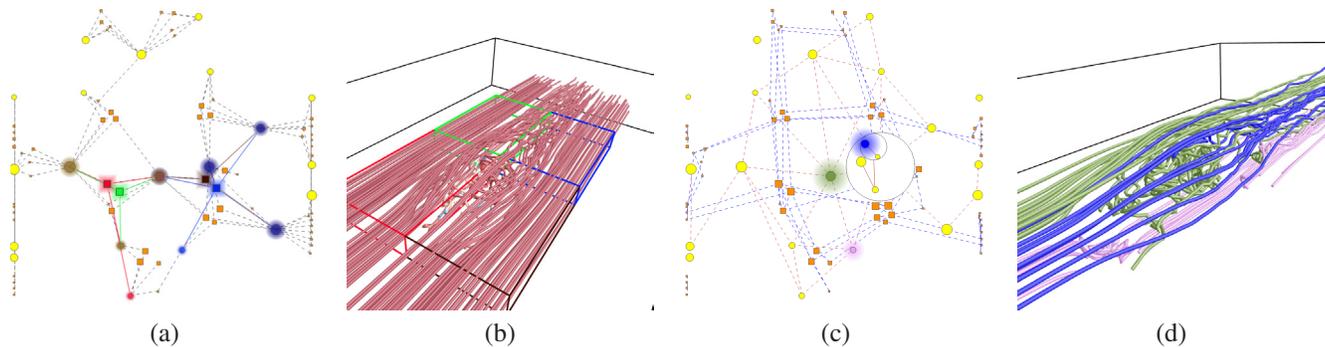


Fig. 13. Exploring the interesting flow pattern in the car flow data set. (a) and (b) show four important R-nodes (shown in red, green, blue and brown) and eight L-nodes that have strong connections to the R-nodes of interest. From these eight L-nodes, (c) and (d) show further selection of three L-nodes (one at the next level of the hierarchy) to capture the main flow structure passing through the car.

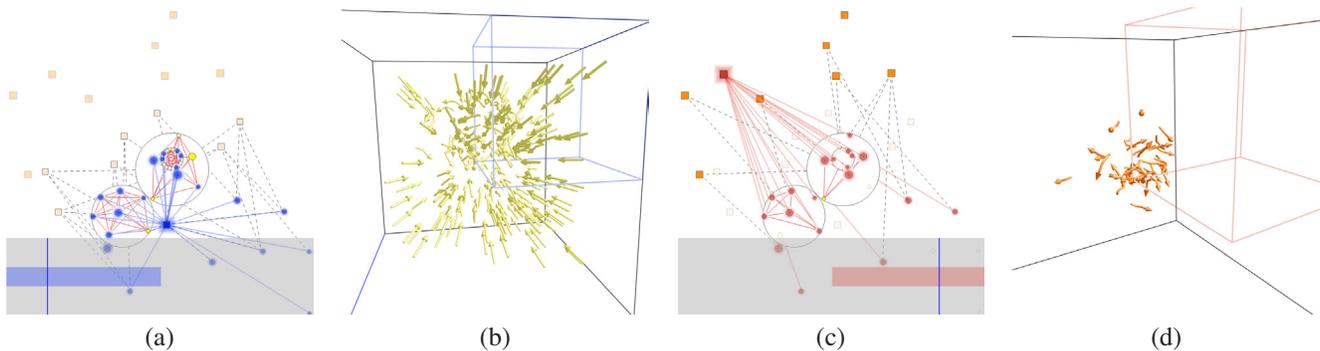


Fig. 14. Exploring two spatiotemporal regions in the unsteady supernova data set. (a) and (c) highlight two R-nodes (shown in blue and red) respectively, and the L-nodes which have strong connections with them. (b) and (d) show the regions and pathlets corresponding to the R-nodes and L-nodes in (a) and (c), respectively. The blue lines in the timeline bars of (a) and (c) indicate the current time steps for pathlet animation in (b) and (d), respectively. The colormap shown in Figure 8 is used to indicate the time steps of the pathlets.

This implies that with the time passing by, some pathline clusters which go through one region in the early time steps may no longer be inside of that region later on. One possible answer to this phenomenon is that there is a sink inside of the region. With the help of our pathlet animation, we can verify the correctness of this assumption. In Figure 14 (b), we show all the pathlets going through the blue R-node in (a). Some pathlets are outside of the blue region because we only show one time step of the animation. Each of these pathlets should go through the region at some specific time step. We can see that most of the pathlets are moving toward the center of the volume. In Figure 14 (d), the spatiotemporal region of the red R-node in (c) is shown. It represents the same spatial region in (b) but covers later time steps. The corresponding pathlets going through this region are also shown. It is clear that many of the pathlets disappear at this time step. Based on this observation, we confirm that there is a sink in the center of the supernova where most of the pathlets are trapped.

Figure 15 gives an example of path comparison for two pathline clusters. The clusters (pathlets) and their corresponding paths are shown in the same color. Figure 15 (b) to (d) show the moving of the pathlets from two clusters as the time evolves. The shared R-nodes are also highlighted in both views with their correspondence labeled by number. From the figure,

we can see that the flow actually follows a circular pattern around the center of the volume according to the order of the labeled spatiotemporal regions (i.e., from 1 to 5).

Case Study 5 — Unsteady Hurricane Data Set. For the unsteady hurricane data set, we increase the number of nodes in the initial graph for detailed exploration by starting the layout from a finer level of node hierarchy. With the help of FlowGraph, we demonstrate how the trajectory of the hurricane center is detected. Figure 16 (a) shows path comparison for three L-nodes in the compound graph and the corresponding pathlines of the selected L-nodes. The R-nodes shared by the three paths are also highlighted in both views. Figure 16 (b), (c) and (d) show the R-node subgraph layouts for three selected time steps, respectively. One interesting finding is that R-nodes in the subgraph are grouped into four well-isolated clusters. Actually, these four clusters form the four horizontal layers along the z dimension of the data set. This indicates that the flows of hurricane almost only move along the xy plane and there is little exchange of flows vertically. In the corresponding three snapshots of pathlet animation, the shared R-nodes are highlighted as the black spatiotemporal regions. We can see that the pathlets follow the hurricane center. With the evolution of time, the trajectory of hurricane follows the order of the shared regions from the

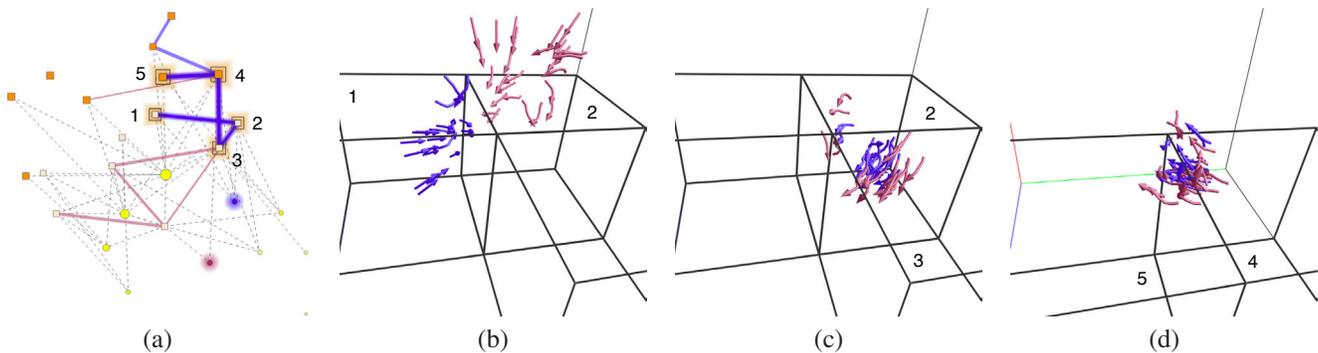


Fig. 15. Path comparison for two pathline clusters in the unsteady supernova data set. (a) shows the two selected L-nodes and their corresponding paths in blue and purple, respectively. Five R-nodes shared in common are marked with 1 to 5. (b) to (d) show three snapshots from pathlet animation where pathlets correspond to the two L-nodes in the same color shown in (a). The corresponding R-nodes passing through in common are marked in (b) to (d). Note that regions 3 and 4 are the same spatial region but cover different time spans.

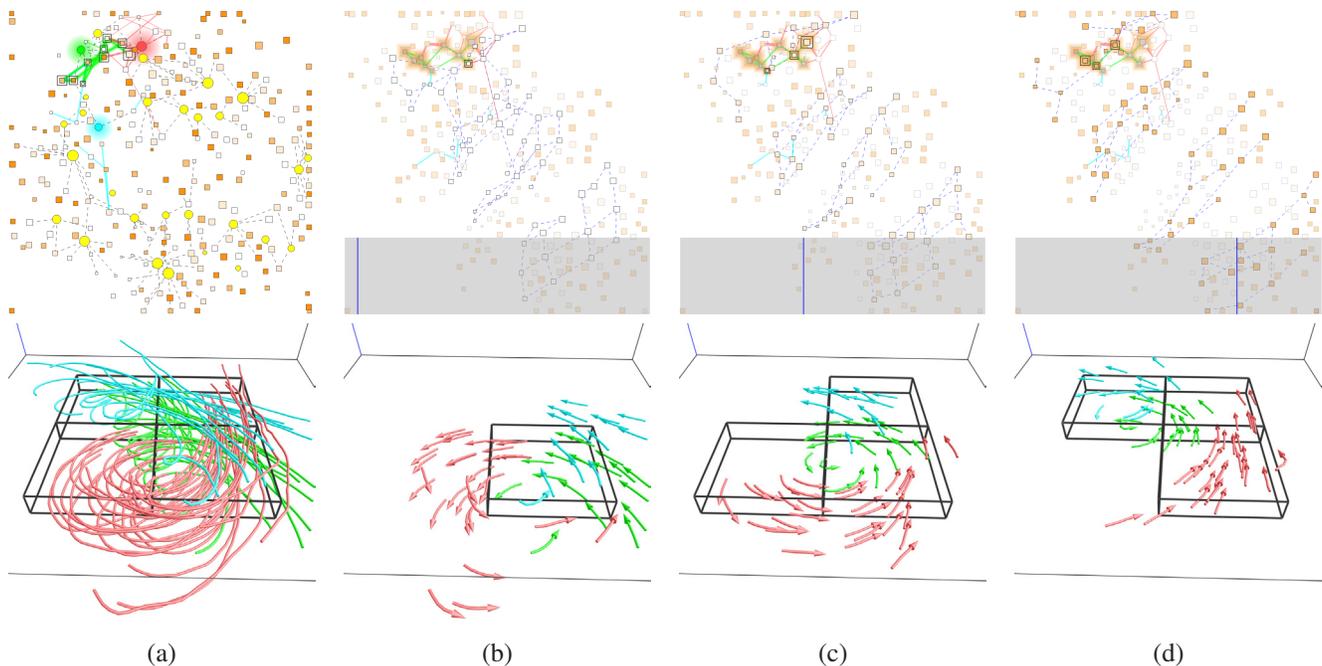


Fig. 16. Path comparison for three pathline clusters in the unsteady hurricane data set. (a) shows three selected L-nodes in green, cyan and red, respectively, and their pathlines. The corresponding paths and shared R-nodes are also shown. These pathlines capture the center of hurricane. (b), (c) and (d) are the R-node subgraph layouts for three different time steps and their corresponding snapshots of pathlet animation. The timeline in each graph indicates the current time step. R-nodes whose time span do not cover the current time step are drawn semitransparently for clear observation. The shared regions in the current time step are also highlighted in the pathline view.

lower-right corner of the volume to the upper-left corner.

8 EMPIRICAL EXPERT EVALUATION

To evaluate the effectiveness of FlowGraph from a practical aspect, we collaborated with a domain expert in biofluids and biomedical engineering, Dr. Jingfeng Jiang. Dr. Jiang's research interests include transforming blood flow characteristics data from raw medical imaging data into clinically-relevant parameters for cardiovascular diseases. The evaluation consists of two major stages. These two stages were completed a few days apart, each took about two hours. In the first stage, basic training was provided to Dr. Jiang. The training aimed at

helping him get familiar with the graphic user interface of the FlowGraph program using several steady and unsteady flow data sets. In the second stage, we provided several other flow data sets for the expert to freely explore by himself. For each data set, Dr. Jiang was instructed to complete one or two pre-determined tasks (such as finding critical points, comparing paths or regions, or identifying flow patterns), assisted by the FlowGraph program. The following is a summary of his feedback.

In general, FlowGraph is a useful and novel tool to explore flow field. It is very helpful in terms of finding the critical patterns within the regions or volumes of interest. For rela-

tively simple flow, the correlation between the graph view and the field line view works well and the connectivity between nodes can show the flow direction clearly. For complex flow fields, adding visual aids will help users quickly understand the graph. For instance, it is feasible to automatically highlight critical regions by streamline density or flow features. The option for viewing L-node or R-node subgraphs provides users a way to explore the flow features in a multiscale fashion. This is particularly useful when one tries to find critical points. The node and edge filtering function allows users to reduce the number of node connections which is crucial for locating important nodes in a complex graph. For unsteady flow field exploration, the spatial constraint of pathlines enables users to visualize self-contained particles by the spatial regions. The temporal constraint of pathlines helps users easily separate slow particles from fast particles if all of them are released from the same position. It could be useful for some biomedical applications. The use of pathlets to represent unsteady flow fields is one important advantage of FlowGraph. First, compared to pathlines, pathlets explicitly show the flow direction information. Second, pathlets are less crowded than pathlines and this helps users observe flow patterns more clearly. Besides, pathlet animation provides a vivid demonstration on the evolution of a flow field over time.

FlowGraph may be directly applied to visualization of cardiovascular flows. The technique will probably work well with the heart and aneurysms. Particularly, the Reynolds number of the physiological flow in the heart is large so that the flow is highly disturbed. Therefore, using streamline and pathline clustering and visualization may help users track hierarchical structures of the flow. The other potential application is to track diffusion flow—another important application in biomedical engineering. Many deliveries of local agents rely on the flow so that more detailed visualization and analysis may help users understand the interactions between the flow (i.e., highway) and the vehicle (i.e., chemical or biological agent). In a clinical setting, clinicians and research engineers often have many images or vector plots to go through. FlowGraph will be very useful in terms of finding the critical patterns, for instance, sources and sinks within the regions of interest. FlowGraph for unsteady flow fields provides the time-resolved information. This can help users identify particle trajectory and therefore infer its residence time, which is relevant to large protein accumulation and subsequent biological effects, e.g., clotting and inflammatory responses. Path comparison will help users study flow mixing. More specifically, if each cluster represents a source of incoming flow, flow paths can visualize how the mixing of flow takes place. Drug delivery will be a good application for this function.

Dr. Jiang also pointed out some possible improvements for FlowGraph. First, a brief user guide will greatly help new users understand and use the program. Second, an “optimized” graph which only shows the critical node connections at the very beginning will help “lazy” users identify important regions easily. Third, for unsteady flow fields, exclusively showing the spatial relationship of two R-nodes which occupy the same spatial region but different time intervals will help users quickly identify such kind of R-nodes and related flow

time-history. Fourth, it would be a good idea to generate graph nodes based on some fundamentals in flow physics, such as sink, source, and vortex core. Finally, if the graph can be arranged based on the flow direction, it will make FlowGraph more intuitive for users to distinguish upstream and downstream flows. We plan to further improve FlowGraph accordingly, making it a more useful tool for domain experts.

9 CONCLUDING REMARKS

We have presented FlowGraph, a new graph-based visual representation that explicitly encodes the field line clusters, spatiotemporal 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 field lines. As we know, 3D field lines normally create dense distribution over the space, which is especially true for large and complex 3D steady and unsteady flow fields. By transforming the field lines, spatiotemporal 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 field line 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 FlowGraph is an abstract representation of the underlying flow field, our objective results and subjective evaluation 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 shall gain a good understanding on how to work with both views effectively to achieve different visual exploration goals. As the size and complexity of flow fields continue to grow, we anticipate future visual analytics systems for flow visualization equipped with such supports comparable to FlowGraph.

ACKNOWLEDGMENTS

This research was supported in part by the U.S. National Science Foundation through grants IIS-1017935, DUE-1105047, CNS-1229297, and IIS-1319363. We would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

- [1] M. J. Berger and J. Olinger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984.
- [2] 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.
- [3] 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.
- [4] 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.

- [5] 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.
- [6] 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.
- [7] M. Edmunds, R. S. Laramee, G. Chen, N. Max, E. Zhang, and C. Ware. Surface based flow visualization. *Computers & Graphics*, 36(8):974–990, 2012.
- [8] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.
- [9] A. L. Fuhrmann and M. E. Gröller. Real-time techniques for 3D flow visualization. In *Proceedings of IEEE Visualization Conference*, pages 305–312, 1998.
- [10] 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.
- [11] 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.
- [12] 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.
- [13] 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.
- [14] 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.
- [15] J. Ma, C. Wang, and C.-K. Shene. FlowGraph: A compound hierarchical graph for flow field exploration. In *Proceedings of IEEE Pacific Visualization Symposium*, pages 233–240, 2013.
- [16] 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.
- [17] 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.
- [18] 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.
- [19] 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.
- [20] Z. Peng and R. S. Laramee. Higher dimensional vector field visualization: A survey. *Theory and Practice of Computer Graphics*, pages 149–163, 2009.
- [21] C. Rössl and H. Theisel. Streamline embedding for 3D vector field exploration. *IEEE Transactions on Visualization and Computer Graphics*, 18(3):407–420, 2012.
- [22] 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.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] A. Telea and J. J. van Wijk. Simplified representation of vector fields. In *Proceedings of IEEE Visualization Conference*, pages 35–42, 1999.
- [27] 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.
- [28] 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.
- [29] X. Ye, D. Kao, and A. Pang. Strategy for seeding 3D streamlines. In *Proceedings of IEEE Visualization Conference*, pages 471–478, 2005.
- [30] 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.



Jun Ma is a PhD student of computer science at Michigan Technological University. His research interests include flow visualization, large-scale data analysis and visualization, and mesh processing. He received a BS degree in computer science from Xidian University, China, in 2006, and a MS degree in computer science from Michigan Technological University in 2009. He is a student member of the IEEE.



Chaoli Wang is an assistant professor of computer science at Michigan Technological University. His research focuses on scientific visualization, large-scale data analysis and visualization, user interface and interaction, information visualization, and visualization in education. He received BE and ME degrees in computer science from Fuzhou University, China, in 1998 and 2001, respectively, and a PhD degree in computer and information science from The Ohio State University in 2006. From 2007 to 2009, he was a postdoctoral researcher at the University of California, Davis. He has served on the program committees of the IEEE SciVis, EuroVis and IEEE PacificVis. He is a member of the IEEE.



Ching-Kuang Shene is a professor of computer science at Michigan Technological University. His research interests include geometric modeling, mesh processing, software visualization, and computer science education. Shene has a PhD degree in computer science from The Johns Hopkins University in 1992, and is a member of ACM, AMS, Eurographics, IEEE/CS, MAA and SIAM.



Jingfeng Jiang is an assistant professor of Biomedical Engineering at Michigan Technological University. Prior to joining MTU, he was a research scientist with Medical Physics Department at University of Wisconsin-Madison. He received his MS in Computer Science and PhD in Civil Engineering from University of Kansas, in 2002 and 2003, respectively. His research and educational interests have been and continue to be in the areas of image-based biomechanics and scientific computing. Since 2006, he has been significantly involved in flow analysis and visualization for the cerebral circulation. He is a member of ASME, AHA and AIUM.