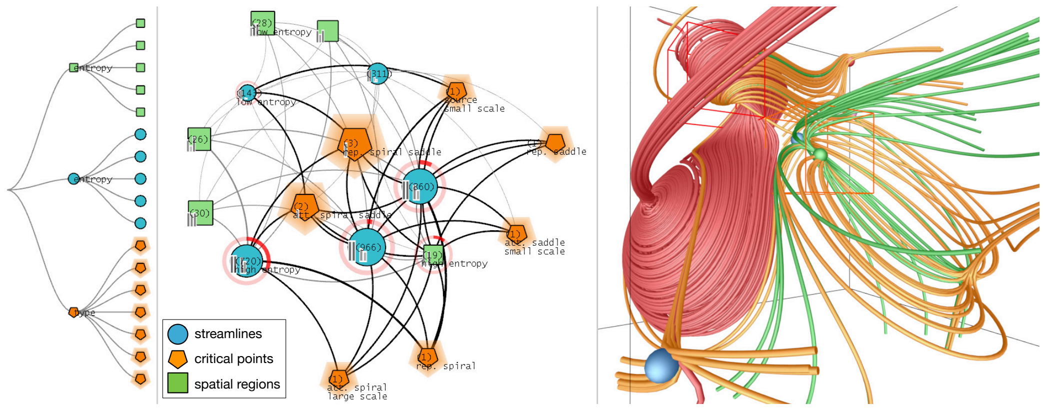
Semantic Flow Graph: A Framework to Explore 3D Flow Fields

Jun Tao*
University of Notre Dame

Chaoli Wang†
University of Notre Dame

Nitesh V. Chawla ‡
University of Notre Dame

Lei Shi §
Chinese Academy of Sciences



(a) (b) (c)

Figure 1: Semantic flow graph (SFG) interface showing the exploration of the five critical points data set. (a) The tree-like history view. (b) The graph view of the current SFG. Selected nodes and their connectors are highlighted by halos and circular percentage bars, respectively. (c) The corresponding volume view.

Abstract

We introduce semantic flow graph (SFG), a novel graph representation and interaction framework that enables users to explore the relationships among key objects (i.e., streamlines, critical points, and spatial regions) of a 3D flow field. The objects and their relationships are organized as a heterogeneous network. We assign each object a set of attributes, based on which a semantic abstraction of the heterogeneous network is generated. We design a suite of operations to explore the underlying flow fields based on this graph representation and abstraction mechanism. Three linked views are developed to display SFG, its node split criteria and history, and the objects in the spatial volume.

1 Motivation

Effectively displaying 3D streamlines faces significant challenges, especially considering the ever-growing size and complexity of flow data generated from scientific simulations. One fundamental challenge is occlusion and clutter, which stems from projecting 3D streamlines to 2D screen. Streamline seeding and selection approaches tackle this challenge by balancing streamline densities among different regions. However, this kind of approaches can only *alleviate* but not *eliminate* the problem, since it is normally impossible to present all flow features clearly at the same time. Some approaches focus on specific types of flow features, for example, critical points. Previous approaches were developed to capture the flow patterns around critical points and reveal their connections. However, they usually serve specific purposes and provide limited interaction to meet various exploration needs. Graph-based approaches provide an abstract representation of the flow field and allow users to interact with them for exploratory visualization. But the existing techniques (e.g., FlowGraph [1]) only consider the spatial relationships among regions and streamlines without explicitly capturing flow features. Furthermore, the graph structures are fixed and still lack of the ability for users to specify different kinds of flow features for exploration.

*e-mail: jtao1@nd.edu
†e-mail: chaoli.wang@nd.edu
‡e-mail: nchawla@nd.edu
§e-mail: shil@ios.ac.cn

2 Our Approach

We present *semantic flow graph* (SFG), a novel solution that leverages semantic graph techniques to explore relationships among streamlines, critical points, and spatial regions. We allow flexible grouping of objects through *dynamic* graph construction and shift our focus to graph exploration. With a rich set of interactions, users are given the unprecedented flexibility to customize the graph according to their own needs in the data exploration and knowledge discovery.

Graph Definition. We construct a heterogeneous graph to capture the relationships among objects in a flow field. We define three types of nodes: *L-nodes* for streamlines, *P-nodes* for critical points, and *R-nodes* for spatial regions. Each type of nodes carries a set of *attributes*, e.g., average curvature and torsion for L-nodes, types of critical points for P-nodes, and entropy of vector directions and magnitudes for R-nodes, etc. Each node is an aggregation of objects that share similar attribute values (e.g., an L-node may represent streamlines with high curvature and high torsion values) or connect to the same set of nodes (e.g., an L-node may represent streamlines connected to a P-node containing all repelling saddles).

To reveal the relationships of these objects, we consider three kinds of edges: L-P edges between L-nodes and P-nodes, L-R edges between L-nodes and R-nodes, and R-P edges between R-nodes and P-nodes. Given an edge $e = <n_i, n_j>$, where n_i and n_j are two nodes connected by e, the weight w_e of e is the summation of weights of connections between objects in n_i and n_j. A streamline and a critical point is connected, if the minimum distance between a point on the streamline and the critical point is smaller than a certain threshold; a streamline and a region is connected, if the streamline passes the region; and a critical point and a region is connected if the region contains the critical point.

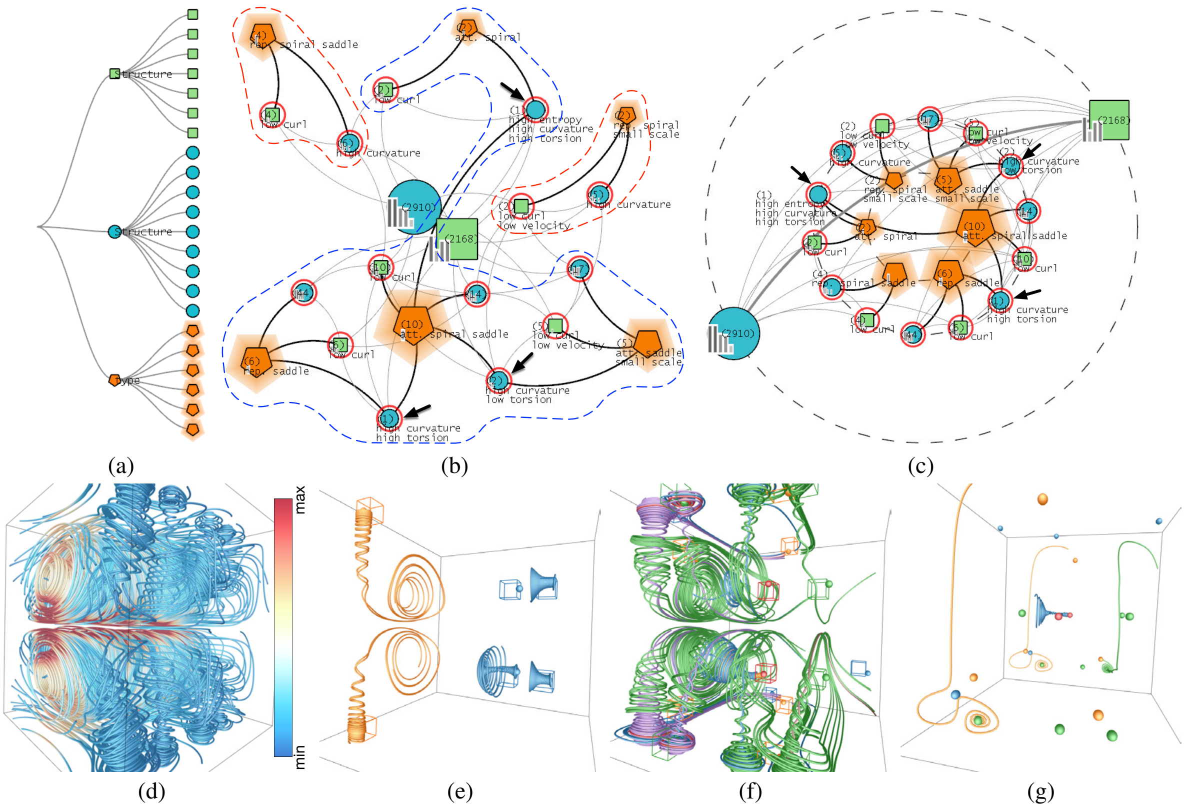


Figure 2: Exploration of the two swirls data set. (a) The node split history. (b) and (c) Neighborhood inspection for all P-nodes without and with the egocentric layout, respectively. (d) 300 streamlines randomly selected from the entire pool of 3000 streamlines. (e) and (f) The objects in the red and blue dashed boundaries of (b), respectively. (g) The connectors of the P-nodes.

Graph Exploration. The exploration of SFG starts with an *ontology graph* containing an L-node representing all streamlines, a P-node representing all critical points, and an R-node representing all spatial regions. Finer structures in the flow field are discovered through splitting the nodes in SFG. We provide two types of split: *attribute-based split* and *structure-based split*. Attribute-based split divides a selected nodes into multiple nodes with a user-specified attribute. Each generated node contains the objects with similar values of the specified attribute. In Figure 2 (b), the P-nodes are generated by performing attribute-based split with "type" attribute, so that each P-node contains critical points of a distinctive type. Structure-based split divides all other nodes based their connection to a selected set of nodes. In Figure 2 (b), the L-nodes and R-nodes are generated using structure-based split with all P-nodes selected, so that each L-node or R-node contains objects connected to the same types of critical points. The node split history is visualized as a tree with labels on parents to indicate the split criteria, as shown in Figure 2 (a).

To investigate node connections, we provide two types of inspection: *neighborhood inspection* and *connector inspection*. Neighborhood inspection reorganizes the nodes according to their connections to a selected set of nodes. In Figure 2 (c), all P-nodes are selected and placed at the center of an egocentric layout. The nodes connected to the P-nodes are placed at the middle layer, and all other nodes are placed at the outer layer. Connector inspection identifies the nodes serving as connectors between any pair of selected nodes. In Figure 1 (b), the connectors of all P-nodes are inspected and highlighted by circular percentage bars. Each percentage bar indicates the percentage of connector objects in the highlighted node. The history view, SFG view, and volume view are synchronized through brushing and linking.

Results. In Figure 1, (b) shows a SFG with L-nodes and R-nodes split by "entropy" and P-nodes split by "type". We perform connector inspection on all P-nodes, and find that the streamlines and regions related to the critical points are normally with high entropy. In (c), the streamlines connecting different types of critical points are displayed.

In Figure 2, (b) shows a SFG with P-nodes split by "type", and L-nodes and R-nodes split according to their connections to the P-nodes by structure-based split. We can observe the overall structure with three groups of nodes: two isolated flow structures are highlighted by the red dashed boundaries, and the corresponding streamlines are shown in (e); the other group is highlighted by the blue dashed boundary and the streamlines are shown in (f). The streamlines connecting different types of critical points are shown in (g). Finally, the neighborhood inspection result with the egocentric layout is shown in (c), and we can see that the neighbors and non-neighbors are clearly separated.

3 Future Directions

We plan to experiment our approach with more data sets and more diversified tasks. Based on the experiments, we would like to derive a set of specific tasks and corresponding operations to performance the tasks. In addition, we will introduce guided exploration into the node split procedure. The system will recommend nodes and attributes to split so that the maximum amount of information can be obtained. Finally, we will explore a more general definition of features instead of critical points and investigate their relationships.

References

[1] 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.