Hierarchical Streamline Bundles for Visualizing 2D Flow Fields

Hongfeng Yu* Sandia National Laboratories
Chaoli Wang† Michigan Tech
Ching-Kuang Shene‡ Michigan Tech
Jacqueline H. Chen§ Sandia National Laboratories

Abstract

We present hierarchical streamline bundles, a new approach to simplifying and visualizing 2D flow fields. Our method first densely seeds a flow field and produces a large number of streamlines that capture important flow features such as critical points. Then, we group spatially neighboring and geometrically similar streamlines to construct a hierarchy from which we extract streamline bundles at different levels of detail. Streamline bundles highlight multiscale flow features and patterns through a clustered yet non-cluttered display. This selective visualization strategy effectively accentuates visual foci and therefore is able to convey the desired insight into the flow fields.

1 Introduction

With respect to flow visualization, Verma et al. [6] proposed three criteria for effective streamline placement and visualization: *coverage*, *uniformity*, and *continuity*. By coverage, it means that no important flow features should be missed and the streamlines should cover the entire domain. By uniformity, it means that the streamlines should be uniformly distributed over the field. By continuity, it means that the streamlines should show flow continuity and therefore long streamlines are preferred.

We agree that capturing all important features such as critical points and revealing flow continuity are essential for generating correct, complete, and pleasing visualization results. Yet one can still produce meaningful visualizations by not covering the entire domain and not adhering to the principle of uniformity. For example, Li et al. [4] demonstrated a technique for succinctly depicting a 2D flow field in an illustrative style using a minimum set of streamlines. For real and complex flow data, prioritizing all flow features enables clear and controllable viewing. Therefore, we conjecture that a suitable solution for flow visualization is to selectively display streamlines that highlight important flow features at various levels of detail (LODs). This paper presents a technique that realizes this idea.

We present *hierarchical streamline bundles*, a new technique for summarizing and visualizing 2D flow fields. Given an input flow field, our method first generates a set of streamlines that captures important flow features through dense seeding. We then cluster streamlines by grouping spatially neighboring and geometrically similar streamlines in a hierarchical manner. Hierarchical streamline bundles are created at different LODs from streamlines that lie on the boundaries of neighboring clusters.

Our work is inspired by fiber clustering in diffusion tensor imaging (DTI) visualization [5] and edge bundling in tree and graph visualization [2]. Clustering neighboring fibers traced from DTI data allows clear observation of the fiber structure and patterns. For tree and graph data, creating bundles from adjacency edges significantly increases the readability of the tree or graph being visualized. Both methods share the common theme of reducing visual clutter and facilitating data understanding. In our scenario, streamline bundles extracted are able to capture prominent flow features in a visually-striking way.

*e-mail: hyu@sandia.gov
†e-mail: chaoliw@mtu.edu
‡e-mail: shene@mtu.edu
§e-mail: jhchen@sandia.gov

2 Our Approach

2.1 Streamline Generation

To ensure that flow features are well captured, our approach first generates streamlines through dense seeding. In practice, we may place seeds randomly in the given 2D flow field until every grid cell in the field has been covered by at least a streamline previously generated. To favor long streamlines, we integrate streamlines as long as possible until they leave the domain, reach critical points, or generate a loop. If a candidate seed is on any of the streamlines previously placed, then we discard this seed as well.

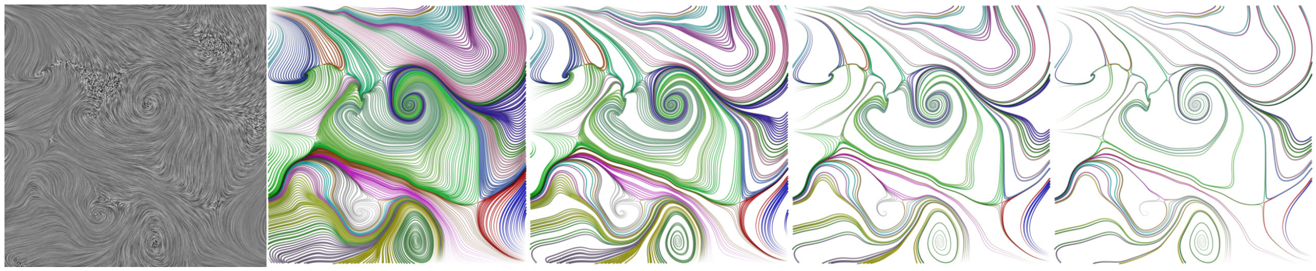
2.2 Hierarchical Clustering

Similarity Measure. With dense seeding, we produce a set of streamlines F with each represented by a set of 2D points p_k, i.e., $F = \{F_i \mid F_i = \{p_k\}\}$. The similarity measure can be defined using the Euclidean distance between pairs of points on two input streamlines F_i and F_j. For example, we can form point pairs by mapping each point of one streamline to the *closest* point on the other streamline [1]. Three pairwise distances between F_i and F_j can be used: the closest point distance, mean of closest point distances, and Hausdorff distance.

Bottom-Up Clustering. To cluster the streamlines, we use an agglomerative hierarchical clustering. This bottom-up method begins with each streamline in a distinct cluster, and successively merges the two most similar clusters together until a stopping criterion is satisfied. Different variations of hierarchical clustering algorithms exist depending on how the similarity between a pair of clusters is defined. Among them, the single-link and complete-link algorithms are most popular ones [3]. In the *single-link* method, the distance between two clusters is the *minimum* of the distances between all pairs of items (one item from the first cluster and the other from the second). In the *complete-link* method, the *maximum* of the distances between all pairs of items is used. In either case, two clusters are merged to form a larger cluster based on minimum distance criteria. Hierarchical algorithms are more versatile than partitional algorithms such as the k-means algorithm. For example, the single-link clustering method performs well on data sets containing non-isotropic clusters, including well-separated, chain-like, and concentric clusters while the k-means clustering algorithm works well only on data sets having isotropic clusters.

Moberts et al. [5] evaluated the combinations of four different clustering methods (single-link, complete-link, weighted-average, and shared nearest neighbor) and four different similarity measures (closest point distance, mean of closest point distances, Hausdorff distance, and end point distance) in the context of DTI fiber clustering. They reported that the use of hierarchical single-link clustering combined with the mean of closest point distances gives the best results. Since DTI fibers and general streamlines share great similarity, we also use the single-link with the mean of closest point distances in our streamline clustering.

During the clustering process, we generate a binary tree to indicate which two clusters are merged at each iteration. Each node in



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

Figure 1: (a) the LIC image of a 2D flow field. (b) a certain LOD of the streamline hierarchy with 37 clusters. (c)-(e) coarsening streamlines at each cluster in the LOD to gradually reveal streamline bundles. 66% and 33% of streamlines in each cluster are displayed in (c) and (d), respectively. (e) only shows streamlines that lie on the boundaries of neighboring clusters.

the tree has a step value, s, recording at which step the associated cluster is created. By default, $s = 0$ for all leaves and $s = N-1$ for the root, where N is the number of input streamlines. For each tree node, we also record the number of streamlines in its cluster.

2.3 Streamline Bundles

Boundary Streamlines. Given a cluster in the hierarchy, there are two ways to define *representative* streamlines and form the bundle. We can either use the streamlines close to the cluster *centroid* or use the streamlines along the cluster *boundary*. Choosing streamlines close to the centroid is commonly used in most cases. However, to form streamline bundles, we choose boundary streamlines to represent the cluster. Our rationale is that streamline bundles should highlight flow features and patterns such as critical points. For a source or sink, it does not matter if we select close-to-centroid or boundary streamlines since either group of streamlines can approach arbitrarily close to the source or sink. However, only boundary streamlines are closest to a saddle and are thus best to reveal the saddle together with other boundary streamlines of neighboring clusters. Therefore, we define *streamline bundles* as the union of streamlines that lie on the boundaries of neighboring clusters. With the agglomerative streamline clustering, we can create hierarchical streamline bundles at various LODs.

Bundle Extraction. To identify boundary streamlines in each cluster, we take a top-down approach starting from two clusters. The streamlines lie on a cluster's boundary can be extracted using a geometry-based approach. We opt for a simpler solution. Our approach first uses a 2D grid, which may have the same resolution as the original 2D flow field, to cover the two clusters. Then for each grid cell, we check if it is intersected by streamlines of different clusters or not. If not, we delete the grid cell from further consideration. Otherwise, we identify streamline intersections on the grid cell's boundary. We sort all these intersections in the clockwise order from which we identify all pairs of consecutive intersections that belong to two boundary streamlines of different clusters. For every pair of intersections, we can identify two boundary streamlines and only save boundary streamlines previously unidentified. We delete all other grid cells that are on the identified boundary streamlines from further consideration. This process continues until all grid cells have been checked. In this way, we can identify all boundary streamlines for the two clusters. This algorithm also works the same for every hierarchical level when we split a cluster into two smaller ones. The only difference is that we can apply a bounding box to quickly rule out grid cells outside of the current streamlines being considered to speed up the extraction. In each level, the two child nodes will nicely inherit the boundary streamlines identified from the parent node as new boundary streamlines are identified. For each cluster that has its boundary streamlines identified, we sort all the streamlines in the cluster in the nondecreasing order based on their minimal distances to all boundary streamlines. This order will be used to control the density of streamline bundles. All the above steps can be performed during the preprocessing stage.

Runtime Visualization. At run time, given a LOD, the streamline bundle is formed by iteratively removing the streamlines in each cluster in the sorted order.[1] The number of streamlines left can be controlled either with a certain percentage or a fixed number. The user can interactively change either parameter and observe an animated effect illustrating how streamlines are removed successively from the clusters to reveal the bundles. In Figure 1, we show the results of hierarchical clustering of streamlines for a flow data set and the gradual coarsening of streamlines in each cluster to reveal streamline bundles. As we can see, streamline bundles shown in Figure 1 (e) succinctly captures the underlying flow features.

3 Conclusions and Future Work

The hierarchical streamline bundles we have introduced offer a new way to characterize and visualize the flow structure and patterns in multiscale fashion. Streamline bundles highlight critical points clearly and concisely. Exploring the hierarchy allows a complete visualization of important flow features. Thanks to selective streamline display and flexible LOD refinement, our multiresolution technique is scalable and is promising for viewing large and complex flow fields. In the future, we would like to seek a cost-effective way to generate streamlines without enforcing the dense seeding condition. We will also extend this approach to handle real-world 3D complex flow fields.

References

[1] I. Corouge, S. Gouttard, and G. Gerig. Towards a shape model of white matter fiber bundles using diffusion tensor MRI. In *Proceedings of International Symposium on Biomedical Imaging*, pages 344–347, 2004.

[2] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[3] A. K. Jain, M. N. Nurty, and P. J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[4] L. Li, H.-H. Hsieh, and H.-W. Shen. Illustrative streamline placement and visualization. In *Proceedings of IEEE VGTC Pacific Visualization Symposium*, pages 79–86, 2008.

[5] 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.

[6] V. Verma, D. Kao, and A. Pang. A flow-guided streamline seeding strategy. In *Proceedings of IEEE Visualization Conference*, pages 163–170, 2000.

[1] It is also feasible to use different distribution functions to control the streamline placement in each cluster with respect to the boundary streamlines. This is not discussed in this paper due to the space limit.