

Integral Curve Clustering and Simplification for Flow Visualization: A Comparative Evaluation

Lieyu Shi, Robert S. Laramee, *Member, IEEE*, and Guoning Chen, *Member, IEEE*

Abstract—Unsupervised clustering techniques have been widely applied to flow simulation data to alleviate clutter and occlusion in the resulting visualization. However, there is an absence of systematic guidelines for users to evaluate (both quantitatively and visually) the appropriate clustering technique and similarity measures for streamline and pathline curves. In this work, we provide an overview of a number of prevailing curve clustering techniques. We then perform a comprehensive experimental study to qualitatively and quantitatively compare these clustering techniques coupled with popular similarity measures used in the flow visualization literature. Based on our experimental results, we derive empirical guidelines for selecting the appropriate clustering technique and similarity measure given the requirements of the visualization task. We believe our work will inform the task of generating meaningful reduced representations for large-scale flow data and inspire the continuous investigation of a more refined guidance on clustering technique selection.

Index Terms—Clustering technique, similarity measures, flow visualization, experimental study, empirical guidelines, quantitative comparisons.



1 INTRODUCTION

VECTOR fields are commonly used in various engineering and scientific applications to help experts understand and assess dynamical systems. Among many visualization techniques, integral curve based approaches are widely employed to visually depict the behavior of complex 3D vector fields (especially, 3D flows) due to their intuitive representation of the flow characteristics via their geometric shapes. One common goal for integral curve-based visualization techniques is to choose a proper set of curves to visually draw the attention of users to important flow regions (with interesting flow patterns) based on their distribution and geometric characteristics (e.g., highly rotating). However, choosing such a set of integral curves is challenging due to the complex nature of the flow behavior that often results in clutter, which is further worsened by 3D occlusion. There are a number of existing techniques to address this challenge, such as evenly-spaced curve placement [1], topology-based integral curve placement [2], [3], information entropy-based integral curve placement and selection [4], [5], [6], and viewpoint-dependent integral curve filtering [7], [8], [9]. This work focuses on the integral curve clustering techniques [10], [11], [12], [13], [14], [15], as it is a general framework that has been proven effective in achieving information reduction by grouping (usually raw input) data with similar characteristics into larger and coarser representations (i.e., clusters and/or segments).

In past years, unsupervised clustering techniques borrowed from the data mining and analytics literature have been applied to flow simulation data to alleviate clutter and occlusion and simplify overall visual representation while highlighting meaningful patterns of the flow. Among all clustering methods listed by Xu and Tian [16], k-means, agglomerative hierarchical clustering (AHC) with various linkages (complete, single, average and Ward's),

spectral clustering (SC), affinity propagation (AP) and DBSCAN are the most popular and dominant techniques for integral curve clustering and abstraction. Incorporated with either customized or well-accepted similarity measures, a specific clustering framework chosen from the aforementioned clustering techniques is able to facilitate an abstraction of specific vector fields while capturing important flow features of interest [10], [11], [13], [17], [18], [19]. Despite many promising clustering results, existing methods still suffer from the following:

- There still lacks well-documented guidance among countless clustering techniques to inform selection of the appropriate combination of similarity measure and clustering method to achieve the desired results. There is no comprehensive study to evaluate different combinations of clustering techniques and similarity measures based on criteria, such as the desired features to capture and the computational complexity.
- How can a user quantitatively and visually evaluate different combinations of clustering algorithms with similarity measures? Whether a *good* evaluation (quantitative) score for a specific clustering combination indicates a *good* visualization of the clustering results for the input streamlines/pathlines still remains an open question.

Even though previous work attempts to quantitatively analyze clustering techniques for blood flow data visualization [10], [11], their analysis of clustering techniques and similarity measures is still incomplete and insufficient to address the aforementioned challenges. To derive guidelines for selecting an informed clustering technique and similarity measure for flow simulation data, we start with a comprehensive experimental study, in which we compare the performance (both qualitatively and quantitatively) of the most popularly used combinations of existing clustering techniques and similarity measures in the flow visualization literature on a selected set of streamline and pathline data sets. From this comprehensive evaluation, we derive empirical guidelines in the form of summary metrics for different combinations of clustering algorithms and similarity measures with a number of flow visualization tasks in practice. We believe this is a valuable contribution to the

- L. Shi and G. Chen are with the Department of Computer Science, University of Houston, Houston, TX, 77004. E-mail: guoning.chen@gmail.com
- R. S. Laramee is an Associate Professor of the Department of Computer Science, Swansea University, SA2 8PP, Wales, UK. Email: R.S.Laramee@swansea.ac.uk

Manuscript received XXXXX XX, XXXX; revised XXXX XX, XXXX.

flow visualization community and will inspire similar follow-up work to refine these guidelines. **We release our complete implementation of the clustering algorithms and similarity measures via Github¹ for reproducibility by others.**

The rest of the paper is organized as follows: Sect. 2 briefly reviews different clustering techniques and the advantages and disadvantages of each. Sect. 3 describes our experimental set-up for the evaluation of clustering algorithms with the selected flow data sets. Sect. 4 contains detailed quantitative and visual comparisons of different clustering techniques with various similarity measures, which leads to empirical guidelines for the selection of an informed clustering technique for flow simulation data. Finally, conclusions and future work are discussed in Sect. 5.

2 OVERVIEW OF CLUSTERING TECHNIQUES

In this section, we provide a concise description of the clustering methods prevalent in flow visualization (see Table 1 for an overview), and discuss their primary advantages and disadvantages based on performance analysis and well-recognized properties [40].

2.1 k-center Clustering

According to Xu and Tian [16] and Xu and Wunsch [41], k-center clustering methods belong to an exclusive, complete and well-separated clustering category without considering noise or outliers. k-center clustering has been applied with MCP [10], [11], end-point distance and sub-bundle properties for illustrative visualization [18].

Depending on whether the center of a cluster is the centroid or the medoid, k-center clustering can be divided into k-means and k-medoids [42] respectively, and the former appears frequently as the baseline for comparison [10], [11], [18] in integral curve clustering. From an optimization perspective, k-means minimizes squared summation of the Euclidean distance, while k-medoids minimizes the Manhattan norm instead. k-means has some well-studied strengths and shortcomings [43].

Pros

- + It benefits from linear complexity in both time $O(knt)$ (k-medoids has $O(k(n-k)^2)$ complexity) and memory w.r.t. the number of the input curves, hence can potentially handle large-scale data.
- + It is easy to implement in parallel.
- + Its objective function can be easily modified or adjusted for more controllable design.
- + It is versatile in clustering candidates not only based on coordinates, but also arbitrary vectors, e.g., self-defined shape property vectors [18].

Cons

- The optimization can get trapped in local minimum and hence can lead to sub-optimal clustering.
- It works best for elliptic and convex cluster shapes (k-means favors spherical clusters while k-medoids is less strict), and sensitive to outliers, noise and often fails in handling non-globular clusters and clusters with widely different sizes [10], [11].
- A preset cluster number k is not easily chosen for optimal clustering, and could be determined by many complicated methods, e.g., joint probability change [44] and the elbow method [45].

- Random initialization can affect the clustering result and hence k-means lacks stability. It can be improved by k-means++ [46], bisection k-means [47] or simply iterating k-means several times and choosing the best result [10], [11].

Remark k-means stands out in computational efficiency and performance even though it suffers from limitations of cluster shape and size, and can still provide a first overview of a data set if combined with a specifically designed similarity measure.

2.2 Hierarchical Clustering

Hierarchical clustering generally falls into two types, agglomerative (bottom-up) and divisive (top-down) [48], and often agglomerative is more practical because divisive methods have $O(2^n)$ complexity. Depending on the linkage criteria, agglomerative hierarchical clustering (AHC) can feature single-linkage, complete-linkage, average-linkage and Ward's method, and can have overall time complexity $O(n^3)$ or reduced to $O(n^2 \log n)$ with a heap.

AHC is widely used in flow visualization. It was firstly applied with point-wise distance metrics, e.g., mean-of-closest-point (MCP), for DTI fiber bundle clustering. Single-linkage AHC with MCP previously achieved the best clustering effect [17], [20], [21], [22], [23], while average-linkage with MCP was better recognized in blood flow clustering [10], [11]. Average-linkage AHC with dimensionality-reduced Euclidean distance by principal component analysis (PCA) was also applied to the clustering challenge for vector field ensembles [15]. Besides, AHC can be coupled with many customized similarity measures that describe either the spatial distance or shape similarity of integral curves. Examples include the average-linkage AHC with a weighted end-curve-distance [12], average-linkage AHC with weighted form of signature-based similarity and mean distance [13], single-linkage AHC with mean-of-thresholded-closest-distance for fiber bundle clustering [24], Wards-variance AHC with segment matching cost distance [27], penalized-linkage AHC with a DTW-based histogram similarity measure [28], average-linkage with the string matching cost measure [14], self-defined AHC with a graph-based similarity measure [29], average-linkage AHC with a specific spatio-temporal similarity measure for adjacent blood flow pattern classification [25], [26].

Selection of linkage is critical for the AHC clustering result, and analysis has been provided previously [10], [11], [21]. In flow visualization, single-linkage [17], [20], [21], [22], [23], [24] and average-linkage [10], [11], [12], [13], [14], [15], [25], [26] are the most frequently employed.

Besides the conventional AHC, a fast and single-scan hierarchical clustering algorithm BIRCH was also proposed based on B+ tree [49], which can handle larger-scale data with higher efficiency than conventional AHC for a given set of resources (e.g., memory). It has complexity of $O(n)$, but favors spherical cluster shapes because the clustering algorithm was derived from variance computation [16]. We have not found BIRCH applied in flow visualization possibly because irregular cluster shapes cannot be detected by BIRCH.

Pros

- + AHC is able to handle clusters of different sizes and arbitrary shapes and can often show hierarchical structures using a dendrogram.
- + An AHC result is usually stable, and can generate any number of clusters once the hierarchical merging tree is built.
- + If the number of clusters is not known in advance, users can interactively browse the cluster hierarchy with only local

1. <https://github.com/lieyushi/FlowCurveClustering>

TABLE 1
References of clustering work in flow visualization (details can be seen in Sect. 2)

Clustering Groups	Algorithms	Paper Reference in Flow Visualization	Complexity		Cluster Shapes	Parameter Input
			Time	Space		
k-center	k-means k-medoids	[10], [11], [18] -	$O(knt)$ $O(k(n-k)^2)$	$O(n+k)$ $O(n+k)$	Spherical Convex	k, t k, t
Agglomerative Hierarchical Clustering	Single-linkage AHC Average-linkage AHC others	[17], [20], [21], [22], [23], [24] [10], [11], [12], [13], [14], [25], [26] [27], [28], [29]	$O(n^2 \log n)$	$O(n^2)$	Arbitrary Convex Convex	k
BIRCH	-	-	$O(n)$	$O(n)$	Spherical	Distance threshold
Density-Based Clustering	DBSCAN OPTICS	[19], [30], [31], [32] -	$O(n \log n)$ $O(n^2)$	$O(n)$ $O(n)$	Arbitrary Arbitrary	ϵ, minPts $\epsilon, \text{minPts}, \xi$
Spectral Clustering	-	[10], [11], [19], [33], [34], [35]	$O(n^\omega)$	$O(n^2)$	Arbitrary	k , scaling factor
Affinity Propagation	-	[36], [37], [38], [39]	$O(n^2 \log n)$	$O(n^2)$	-	$S(i, i)$
PCA-based Clustering	PCA+AHC	[15]	$O(nm^2)$	$O(nm)$	Convex	$ex(r), k$

updates to the result, which are easier to track than global changes as in, e.g., spectral clustering.

Cons

- AHC is expensive in terms of computational and storage requirements, with a time complexity of $O(n^3)$ (or $O(n^2 \log n)$ [16] and space complexity of $O(n^2)$. This severely restricts application of AHC to large-scale data sets.
- AHC lacks a global objective function hence has limited controllability and predictability. The latest work has proposed an objective function for AHC [50] but it is still difficult for practical problems.

Remark The biggest advantage of AHC is its ability to detect clusters of arbitrary shape and size, which is also restricted by computational complexity for practical applications. However, because the input to the clustering algorithms in the existing flow visualization applications is usually not too large, AHC is still regarded as the most prominent and the state-of-the-art clustering algorithm in the flow visualization literature.

2.3 Density-based Clustering

In density-based clustering [51], clusters are defined as areas of higher density than the remainder of the data set, and objects in sparse areas are often considered as noise or border points. The most popular density-based clustering method is DBSCAN [52]. It has been applied in visualization with spatial distance, e.g., weighted sum of local distance [53], segment endpoint distance [19] and with Euclidean distance in low dimensional space (obtained using the feature descriptors for streamlines and stream surfaces through auto-encoder [30]), respectively. DBSCAN is also combined with the averaged centerline distance (ACD) in aortic vortex flow clustering [31] and with a modified coherence distance encoding FTLE values for blood flow exploration [32].

DBSCAN requires two parameters, ϵ (radius) and minPts (minimum number of points). It is a query-based algorithm to detect candidates as either core, border or noise based on their neighboring connected points. The runtime complexity of DBSCAN can be reduced to $O(n \log n)$ with a naive implementation as $O(n^2)$, and requires memory $O(n^2)$ for an implementation that needs to store the distance matrix and $O(n \log n)$ for an implementation without storing the matrix.

Besides DBSCAN, another popular density-based clustering method is OPTICS [54], which is similar to DBSCAN but addresses one of the limitations of DBSCAN, that is, DBSCAN fails to detect meaningful clusters of varying density. Compared to DBSCAN, OPTICS replaces the ϵ with a maximum value that

affects performance, and minPts specifies the minimum neighbor size to find, hence it lowers the difficulty of parameterization.

Pros

- + It can detect noise and is robust to outliers.
- + It can find arbitrarily shaped clusters and does not require an *a priori* preset cluster number.

Cons

- Parameter setting is very difficult for a data set without *a priori* knowledge, and the final result is quite sensitive to parameter settings.
- Parallelism of density-based clustering is challenging due to the data dependency of connectivity expansion. Recent parallel implementations on distributed systems have been proposed [55], [56], [57], but they are yet to be applied to the clustering problems in visualization.

Remark The most promising aspect of density-based clustering is that it is robust to outliers and noise, and able to detect arbitrarily shaped clusters with lower cost than AHC, while parameter setting might require prior knowledge of the data.

2.4 Spectral Clustering

Spectral clustering (SC) [58] is a classical clustering method utilizing a spectrum of data similarity matrix to perform dimension reduction before clustering in a lower dimensional space. From the dimension-reduction point of view, SC benefits from the same intuition as PCA-based clustering [15]. The difference is that, PCA [15] focuses on the reduction of a coordinate matrix while SC on reduction of a distance matrix. We refer the interested readers to the survey by Luxburg [59] for a detailed description of SC in both theory and practice.

SC has been applied to integral curve clustering, e.g., with an average distance between pairs of nearest points in white matter fiber tract clustering [33], with a user-specified spatial distance in medical image analysis [34], with MCP [10], [11], Hausdorff distance [35] and minimal closest point distance [19], respectively. It has been reported that SC with MCP has achieved better results in blood flow clustering than k-means and AHC [10], [11]. In terms of implementation, the technique presented in [33], [35] uses k-way normalized cut as post eigen-decomposition as by Shi and Malik [58], [19] uses k-means by Ng et al. [60], while [10], [11], [19] uses eigen-rotation minimization [61].

Complexity analysis for SC can be non-trivial due to the fact that we do not know yet whether the sparsity of the normalized Graph Laplacian L impacts the result or not. For time complexity, in addition to the distance matrix computation with $O(n^2)$ complexity,

the complexity of the eigen-decomposition for a non-sparse matrix is $O(n^\omega)$ ($2 < \omega < 2.376$) [62] in the optimal scenario but is $O(n^3)$ in general. Memory complexity is around $O(n^2)$ because several $n \times n$ matrices need to be stored.

Pros

- + SC provides more intrinsic partitioning/segmentation, and can achieve meaningful projection into a lower-dimensional space [35].
- + SC with eigen-rotation can automatically find the optimal number of clusters among a user-given range [10], [11], [19].
- + SC can detect clusters with arbitrary shapes.
- + SC has objective functions, and is versatile for modification.

Cons

- Runtime and memory complexity restricts SC from being applied to large scale data sets.
- SC tends to form clusters of even size and might fail for clusters with varying sizes.

Remark Spectral clustering gradually gains attention via its application to blood flow visualization, and may provide more natural and meaningful cluster extraction at the cost of higher runtime and memory usage.

2.5 Affinity Propagation

Affinity propagation (AP) is a clustering algorithm based on the concept of “message passing” between data points [63]. It finds “exemplars” for each cluster, and it does not require the number of clusters as user input.

AP has been used in flow visualization literature mostly with shape similarity measures of streamlines, pattern-based distance metrics, e.g., bag-of-features from feature vector computation of streamline signatures [36], [37] and the adapted Procrustes distance on re-sampled points [38], [39]. The initial values of $S(i, i)$ which are diagonal entries of S are vital in controlling clusters the algorithm produces. According to Frey and Dueck [63], $S(i, i)$ is typically initialized to the median similarity of all pairs of inputs.

Pros

- + AP is simple and easy to implement in parallel.
- + AP is insensitive to outliers and noise.
- + AP does not require user input of cluster numbers and it automatically detects a suitable number of clusters based on the given parameters.

Cons

- The runtime complexity is $O(n^2 \log n)$ and memory usage is $O(n^2)$ [16], which may restrict its application to large data.
- The clustering result is sensitive to the parameters involved in the AP algorithm, e.g., diagonal entry initialization and coefficient in updating responsibility and availability matrix.

Remark AP has even more expensive computational cost (i.e., time complexity) than AHC, and the final result is highly dependent on initialization. Right now AP is only combined with very specifically-designed similarity measures, and sometimes the clustering result is very difficult to interpret.

3 EXPERIMENTAL SETUP

In this section, we describe our setup for the experimental study and evaluation of the aforementioned clustering algorithms using a number of well-known flow simulation data sets.

3.1 Flow Simulation Data Sets

We place streamlines densely in the benchmark flow simulation behind a square cylinder [35], Bernard convection simulation [64], crayfish [38], tornado [38], hurricane Isabel simulation [17], and solar plume [38], respectively. All streamlines are generated using a uniform sampling strategy. Short-length streamlines are preserved in case they convey important flow information. We perform pathline tracing for three unsteady data sets, i.e., 3D flow behind a square cylinder [65], a tube simulation [66] and blood flow simulation from Berg et al. [67] and Janiga et al. [68].

3.2 Clustering Algorithm Implementation and Parameter Selection

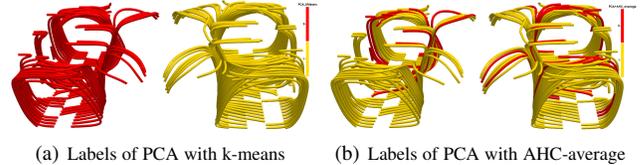


Fig. 1. PCA-k-means shows better and more consistent labels in clustering tested streamline data sets than PCA-AHC [15], e.g., in reduced Bernard. Note that PCA-based clustering resembles direct clustering based on Euclidean distance, and streamlines should be separated into left and right bundles if preset clusters are 2 for this data. PCA-k-means can achieve more consistent labeling than PCA-AHC, so in our experimental study we prefer PCA-k-means over PCA-AHC.

Given the flow simulation data sets in Sect. 3.1, we implement the following clustering algorithms mentioned in Sect. 2. All the clustering algorithms are run on a PC and a cluster. The PC has an Intel Xeon (R) CPU running at 2.40GHz, 32 GB main memory, and an nVidia Quadro K4200 graphics card with 4GB graphics memory. The cluster has 8 nodes and each node uses an Intel Xeon(R) CPU E5-2620 v4 @ 2.10GHz and 64 GB main memory.

- 1) **k-means**: The maximal iteration is set to 50, and k-means++ [46] is provided as an option to the user for better centroid initialization.
- 2) **k-medoids**: We use Eq. 1 of Weiszfeld’s algorithm [69] for an iterative medoid computing

$$m_i^{(t+1)} = \left(\sum_{j=1}^{|S_i|} \frac{x_j}{d(x_j, m_i^{(t)})} \right) / \left(\sum_{j=1}^{|S_i|} \frac{1}{d(x_j, m_i^{(t)})} \right) \quad (1)$$

because it matches the representative extraction of closest and furthest to centroid (i.e., medoids in k-medoids) in Sect. 3.6.

- 3) **PCA clustering**: Our implementation features two differences from Ferstl et al. [15]. First, we use k-means instead of AHC in dimension-reduction space because we find PCA-k-means works better than PCA-AHC in most tested data sets as shown in Fig. 1. Second, we extract the centroid instead of median to make the representative consistent with other clustering results in Sect. 3.6.
- 4) **AHC**: We build a bottom-up dendrogram, i.e., the hierarchical merging tree, until the preset number of clusters is obtained. Since only average and single linkage of AHC are utilized in related work, we only evaluate the results of these two linkages and leave the other two as user options. We adopt the idea from Walter et al. [70] to achieve a faster version of AHC by merging necessary tree nodes beforehand.
- 5) **BIRCH**: BIRCH in Sect. 2.2 is also implemented because of its faster computation of hierarchical clustering. However,

BIRCH requires a distance value as user input which hierarchically merges any two objects within this range. The value selection is intrinsically challenging and it is difficult to achieve similar cluster numbers to other hierarchical clustering methods. To address this challenge, we use a binary-search algorithm to adaptively adjust a distance threshold to obtain roughly similar numbers of clusters until a given maximal number of iterations is reached.

- 6) **DBSCAN**: Parameter setting is challenging and requires prior knowledge of the data sets for DBSCAN. To alleviate this, we provide two methods of setting ϵ (radius). One is to set ϵ to the average of $minPts$ -th smallest distance on all candidates, and the other relies on user input, of which the former is adopted for assessing the DBSCAN clustering due to its simplicity.
- 7) **OPTICS**: Besides setting parameters ϵ and $minPts$, OPTICS is also parameter-sensitive to find clusters because final clusters are determined by valleys of the 2D reachability-plot. We used the naive idea that detects valleys by steepness as in Ankerst et al. [54] to determine the resulting number of clusters.
- 8) **SC**: Similar to Oeltze et al. [10], [11], [19] and Rossl and Theisel [35], we implement both k-means and eigen-rotation minimization for SC, and set the dimension of reduced eigenvector (k) to the preset cluster number. We use 5% as the scaling factor as suggested by Oeltze et al. [10], [11].
- 9) **AP**: The maximal iteration is set to 40, and we use implicit iteration to update availability and responsibility matrix to ensure numerical stability. Similar to Tao et al. [38], [39], we use the minimum similarity values as the preference initialization and two-level clustering to obtain a reduced number of clusters.

3.3 Similarity Measures

It is not possible or technically practical to consider every similarity measure proposed for integral curve comparison, especially those specifically designed for pattern search, which requires complicated pre-processing or segmentation. Therefore, we choose well-known and widely-accepted measures for our clustering study. Additionally, since our clustering analysis for both streamline and pathline data sets is based on geometric properties, we also consider one similarity measure specified only for pathlines (d_T in Meuschke et al. [25], [26]).

- 1) Euclidean distance $d_E(\cdot, \cdot)$ that computes the pairwise Euclidean distance of two curves.
- 2) Fraction norm $d_F(\cdot, \cdot)$, taken from Aggarwal et al. [71] which addresses the *curse of dimensionality* challenge for high-dimensional space. We set $p = 0.5$.
- 3) Geometric similarity measure $d_G(\cdot, \cdot)$, introduced by Shi and Chen [72] based on the intuition that two curves are considered similar if their pairs of piece-wise line segments are parallel.
- 4) Accumulated rotation difference $d_R(\cdot, \cdot)$, which measures the difference of summation of discrete curvature along two curves [73].
- 5) Mean-of-closest-point (MCP) $d_M(\cdot, \cdot)$, is considered a state-of-the-art distance metric in [10], [11], [17], [20], [21], [22], [23].
- 6) Hausdorff distance $d_H(\cdot, \cdot)$ from Rossl and Theisel [35] which is topologically meaningful and forms a metric space.
- 7) Signature-based measure $d_S(\cdot, \cdot)$, proposed by McLoughlin et al. [13], which uses a combination of both closest-point-

distance and χ^2 test of streamline signatures (discrete curvature). We chose a fixed number of signature bins for each streamline/pathline since our sampling strategy can make streamlines/pathlines all equal size, and set $\alpha = 0.5$.

- 8) Adapted Procrustes distance $d_P(\cdot, \cdot)$, used in Tao et al. [38], [39], is defined as the Euclidean distance after Procrustes superimposition [74]. We set the local shape size r to be 7 as Tao et al. [38], [39] and compute the average Procrustes distance for all pairs of local shapes among integral curve pairs.
- 9) Time-series MCP $d_T(\cdot, \cdot)$, introduced in Meuschke et al. [25], [26] which is MCP considering time interval overlapping and mismatching. This similarity measure is specifically suited for temporal pathline similarity computation. Note that since d_T is introduced specifically for pathlines, we will only apply it to our pathline data sets.

In all, d_E , d_F , d_M , d_H and d_T are *spatial similarity measures* that characterize spatial proximity of integral lines, while d_G , d_R and d_P are *shape-based measures* that characterize shape similarity, and d_S belongs to both groups. Note that, we provide a simplified implementation of d_S and d_P using our consistent sampling strategy, and we opt for not discussing the parameters for the distance calculation since our focus is the clustering analysis and evaluation. More details on the formula and discussion of these similarity measures can be found in the supplementary materials.

Note that although most of the above similarity measures are for streamlines (except d_T), they can be applied to pathline comparison as well despite that pathlines may intersect. This is because both streamlines and pathlines are 3D curves with similar geometric characteristics except that for comparing pathlines, points on different pathlines are aligned based on their respective time stamps.

3.4 Sampling Strategy

Since the centroid computation of each cluster in Sect. 3.6 requires all curves to have the same number of samples, we need to re-sample the input curves as a pre-processing step. Three common equal-size sampling strategies are considered.

- **Directly repeating** the whole array with the last vertex coordinates [15]. This strategy is easy to implement but can be problematic for curves with too few vertices.
- **Evenly-spaced re-sampling** given the total number of samples [28]. This strategy requires a scan of each streamline at least twice and often distorts the original streamlines morphologically (e.g., tortuous streamlines might become flat) even with very dense sampling.
- **Sampling based on signatures** (e.g., curvature, torsion or tortuosity) [13], [38], [39]. This strategy is computationally expensive and the segmentation is often sensitive to parameters and user-input thresholds.

Our PCA clustering utilizes direct repeating as in Ferstl et al. [15]. For pathlines, we repeat their starting and/or end points such that all pathlines have the same number of points, including the same starting and terminating times and well-aligned intermediate points in time, which guarantees the cluster centroid of pathlines is temporally meaningful. For streamlines, to achieve both the efficiency and the preservation of the geometric shape of curves as much as possible, we use a sampling method which preserves the original vertices while adding samples to the existing line segments. The total number of samples for each curve is set to be the maximal

vertex number of all input curves to preserve the initial geometry as much as possible, with the cost of additional storage. Given the maximal vertex count as M , for an integral line with n vertices, we uniformly embed $\lfloor \frac{M-n}{n-1} \rfloor$ new samples on each line segment between vertex i and $i+1$ ($1 \leq i \leq n$) starting from the beginning of the integral line, until the maximal vertex count (M) is reached.

3.5 A Priori Cluster Numbers

As discussed in Sect. 2, some clustering algorithms (i.e., k-means, k-medoids, AHC, PCA, SC+k-means) rely on specific number of clusters as input, and users may have no prior knowledge of the simulation. Most work in flow visualization that requires cluster number as input regards cluster number as part of the interactive exploration, e.g., [15], [18], [20], [22]. Optimal cluster numbers could be detected under given k (maximal cluster number) by SC with eigen-rotation minimization as discussed in [10], [11], [19] and SC with k-way cut in [35], which provides a practical way to determine number of clusters if cluster number is needed. The L-method is applied to detect optimal cluster number (less than 20) for blood flow pattern classification [25], [26].

In our paper, we implement two methods to detect an optimal number of clusters for a given streamline/pathline data set. The first one is SC with eigen-rotation minimization [10], [11], [19]. We set k (maximal cluster number) to be a fixed number (i.e., 100 as a compromise between efficiency and accuracy because 20 used in [10], [11], [19] may not be enough), such that eigen-rotation minimization can find the optimal cluster number within this range. The second is the L-method [75] that can find optimal k for hierarchical clustering by iteratively refining the knee of the “cluster-number vs. merged-distance” graph. SC eigen-rotation minimization is claimed to be better than the L-method for blood flow data in [10], [11].

We consider the optimal k obtained using both the SC with eigen-rotation and the L-method for our quantitative study. Our experiments show that neither method is better than the other (see discussion in Sect. 2.1 in the supplementary document). For the flow abstraction obtained using the clustering results, we choose an appropriate cluster number for each data set that is not dependent on either method. This is because too few clusters may fail to capture important features in streamlines/pathlines, and both L-method and SC can sometimes generate very small cluster numbers.

3.6 Select Cluster Representatives

Selecting the representative curves for a reduced representation of the original data is also critical after clustering. A naive method is to choose the centroid (or average) streamline [13], [14] as the representative for each cluster, which is often not adopted because the centroid streamline is artificially generated by averaging the streamlines within the same cluster, thus cannot reveal the authentic flow characteristics. In general, streamlines closest to centroids [18] or streamlines furthest away from centroids [76] are selected because they can often depict certain flow patterns. Besides, when streamlines are projected into lower dimensional space and further clustered, representative streamlines could be chosen as the actual median of clusters in the lower dimensional space [15] or closest to centroid in the streamline-embedding space [35]. Density-based representation is also applied [10], [11], [19] based on the information contained in voxels that streamlines pass through. An attribute-based representative strategy is recommended if the clustering is guided by an attribute of

integral curves [10], [11]. Representative streamlines can also be chosen by iteratively removing the most similar streamlines until characteristic candidates remain [13], or derived from skeleton of line predicate-based streamline bundles [77]. For trajectory clusters, average coordinates w.r.t. average direction vector are collected as representatives [53]. Recently a new representative selection approach based on functional decomposition is proposed and is able to reduce clutter and to find important patterns [78].

Since the focus of this work is on the quality of different combinations of clustering techniques and similarity measures, we intentionally choose a naive strategy to avoid the discussion on how different similarity measures affect the definition of “most-representative”. In particular, we extract the closest or furthest (or called boundary lines by Yu et al. [17]) curve to the centroid as the representative of each cluster. Additionally, we choose stream tubes as a representative visualization suggested by Oeltze et al. [10], [11] because streamtapes introduced by Chen et al. [18] cannot handle densely-distributed vortex rings.

3.7 Clustering Evaluation

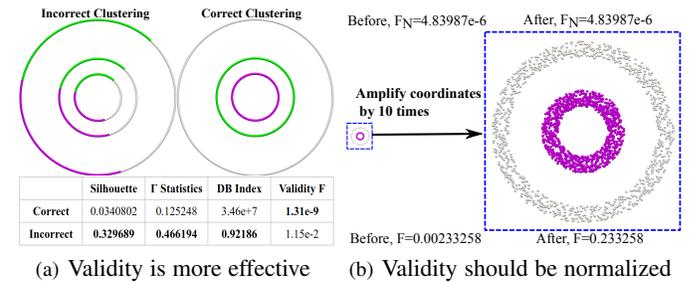


Fig. 2. Validity measurement [79] is a better clustering evaluation metric for arbitrary cluster shapes, and normalized as in Eq. 3 for comparing different similarity measures. The three-ring data set in (a) indicates that validity measurement is a better evaluation metric than silhouette, db index and Γ statistics for non-convex shapes of clusters in point cloud data sets using Euclidean distance (better values marked bold). In (b) we amplify the coordinates of the points by 10 times while maintaining the same clustering, and validity measurement F in Eq. 2 [79] indicates that the clustering result before amplification is better than after, while the normalized validity measurement F_N in Eq. 3 indicates that both clustering results are quantitatively the same.

So far, most integral curve clustering techniques for flow data rely on visual inspection and comparison to assess the clustering quality, which is rather subjective. Despite many quantitative metrics having been introduced to evaluate the clustering quality [41], [80], clustering analysis (especially quantitative analysis) for flow visualization is not very popular to the best of our knowledge. A new weighted normalized adjusted random (WNAR) was introduced in flow visualization for evaluating and validating clustering results by Mobergs et al. [21], which is an external quality measure w.r.t. ground truth. Normalized information distance (NID) borrowed from Vinh et al. [81] is also used to compare bottom-up clustering techniques before and after top-down balancing by Yu et al. [17] and it demonstrates an improvement over the adjusted Rand index. Silhouette width, connectivity, Hubert’s Γ statistic and stability have been used for comparing k-means, AHC and spectral clustering (SC) in [10], [11].

Due to the lack of ground-truth or pre-identified labels for comparison, external evaluation measures are not applicable in our clustering evaluation. We apply the following clustering evaluation

measurements in our experiments partially borrowed from blood flow visualization analysis work [10], [11].

- Silhouette width: According to Oeltze et al. [10], [11], silhouette width is a non-linear combination measure of cluster cohesion and separation with higher value indicating higher cohesion within clusters and good separation among clusters.
- Hubert's Γ statistic: Taken from [10], [11] and implementation by Marghescu [82]. Higher correlation values indicate better equality of clustering results.
- Davies-Bouldin index [83]: The smallest DB value is considered the best, since algorithms that produce clusters with low intra-cluster distances and high inter-cluster distances will have a low DB index.
- Normalized validity measurement: Validity measurement is first proposed by Yousri et al. [79] and described as:

$$F = \min_c \{f_c = h(DD_c) \cdot g(S_c)\} \quad (2)$$

where DD_c measures the distance homogeneity and S_c measures the density separateness for a cluster c , and smaller F indicates better clustering as claimed by Yousri et al. [79]. This novel density-based validity measurement was able to evaluate clustering analysis of arbitrary cluster shapes and densities as in Fig. 2(a) via a minimal spanning tree (MST) algorithm. However, in our experiment we find validity measurement simply cannot evaluate clustering results across different similarity measures because the $h(\cdot, \cdot)$ and $g(\cdot, \cdot)$ in Eq. 2 relate to standard deviation of distance values. Hence we normalize the validity measurement to obtain:

$$F_N = \frac{F}{(d_{max} - d_{min})^2} \quad (3)$$

and show its correctness and effectiveness in Fig. 2(b). Note that only after normalization can the validity measure be applied to quantitatively evaluate and compare the clustering results across various similarity measures.

Even though Yousri et al. [79] indicate validity measurement is a more robust and effective evaluation than conventions (e.g., silhouette, Γ statistics, DB index, all of which only work for convex cluster shapes) for clusters of arbitrary shapes (including clusters with convex shape) and density, we still follow the evaluation work of [10], [11] and consider all these evaluation metrics in our experimental study. The essential reason is that streamlines/pathlines often involve customized similarity measures (some are even not rigorously norm or metric) and point-based knowledge under Euclidean distance could not be simply and directly extended to flow visualization. This might explain why silhouette, e.t.c., are still used as a *Golden Rule* to quantitatively evaluate among different flow clustering techniques as in [10], [11].

4 EXPERIMENTAL RESULTS AND DISCUSSION

In this section, we report our experimental results and analysis. The average computation time of clustering algorithms with similarity measures for streamlines/pathlines is listed in Table 3 of the supplementary material, and the time includes distance matrix computation because all the evaluation metrics in Sect. 3.7 depend on pair-wise similarity values between integral curves. From the table we see that two density-based clustering algorithms, i.e., DBSCAN and OPTICS, have the smallest computational overhead, while SC with eigen-rotation and AP take the most computational

time. Meanwhile for similarity measures, d_E and d_R cost the least while d_M and d_H cost the most.

We also compare the clustering results both quantitatively (Sect. 4.1) and qualitatively (Sect. 4.2). From these results, we summarize a set of general guidelines for the selection of optimal clustering technique and similarity measures for specific tasks (Sect. 4.3).

4.1 Comprehensive Quantitative Analysis

We run the clustering algorithms of Sect. 3.2 with similarity measures listed in Sect. 3.3 on 8 streamline/pathline data sets (see Sect. 3.1). As discussed in Sect. 3.5, we adopt both optimal cluster numbers by SC with eigen-rotation and the L-method if clustering algorithms require the cluster number as input and average the quality values of the two optimal cluster numbers for each evaluation metric. The detailed tables of evaluation metrics for each data set can be found in the supplementary document. Afterwards, we calculate the average evaluation separately over all streamline and pathline data sets, respectively, and create two evaluation metric tables (Table 13 for average of streamlines and Table 14 for pathlines). Due to space limits we place all evaluation tables in the supplementary material.

We use a **ranking-based** visualization technique similar to what has been applied in hex-mesh quality visualization in [84] for two evaluation matrices. Firstly, we map all four evaluation matrices to the range $[0.1, 1.0]$ so that 0.1 always denotes the worst value and 1.0 for the best, e.g., largest silhouette value and smallest validity marked as 1.0. The reason for starting from 0.1 rather than 0 is because evaluation metrics for some clustering combinations do not have values. To avoid mapping both those non-existing values and the worst value to zero, we map only the non-existing values to zero, while the worst value to 0.1. In particular, we use dynamic mapping for DB index and normalize validity measures since their values vary significantly.

Fig. 3 shows the averaged evaluation/quality values for the individual combinations of clustering techniques and similarity measures organized w.r.t. the four evaluation metrics in the form of four matrices for streamlines. These four matrices summarize the evaluation measures obtained from the six streamline data sets. To help us better compare across different combinations, we re-order the rows and columns of the four matrices. Specifically, for each row, we compute the average value that characterizes the overall quality of a clustering technique w.r.t. all similarity measures. We then sort the rows based on their respective values in descending order from top to bottom so that the top row corresponds to the clustering technique that has the best average value. Similarly, we compute the average value for each column to characterize the corresponding similarity measure, based on which we sort the columns from left to right so that the similarity that works the best with all clustering techniques is in the left-most column. Note that PCA clustering is only performed based on d_E (Euclidean distance) on lower-dimension space, so all the other similarity measures can not be combined with PCA.

4.1.1 Results of Streamline and Pathline Clustering

Since clustering techniques and similarity measures are ranked across four evaluation metrics, we calculate the average ranking score of each clustering technique and similarity measure for streamlines and pathlines, and sort clustering techniques and similarity measures by the average ranking score, respectively.

Streamlines: From Fig. 3 we see that for **streamlines the top three clustering algorithms with a highest average ranking**

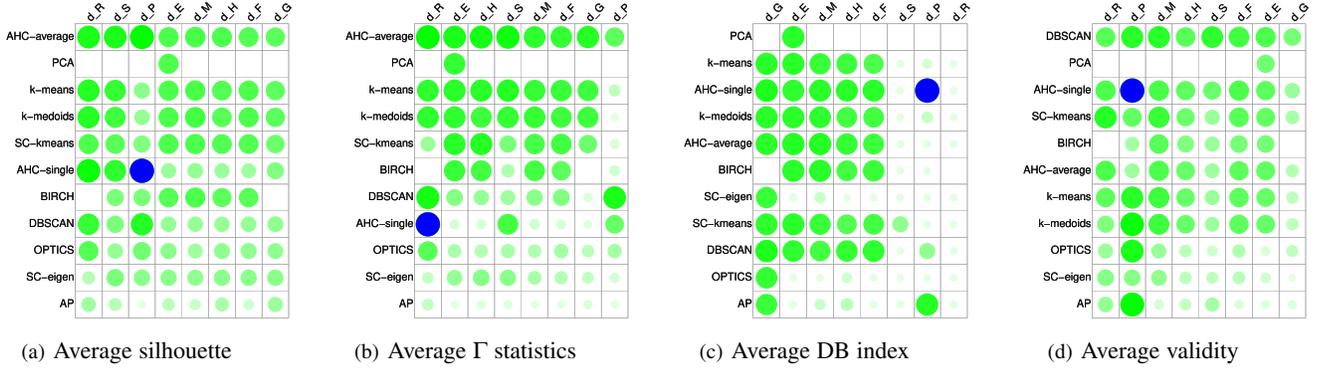


Fig. 3. Ranking visualization of evaluation metrics based on average evaluations of streamline data sets (best cell highlighted in blue). We can observe that PCA, AHC-average and k-means always result in a better silhouette, Γ statistics and DB index, while validity favors DBSCAN. Overall, PCA (1.75 on average) and d_R (average 2.75) respectively have the best evaluation result across all four quality metrics.

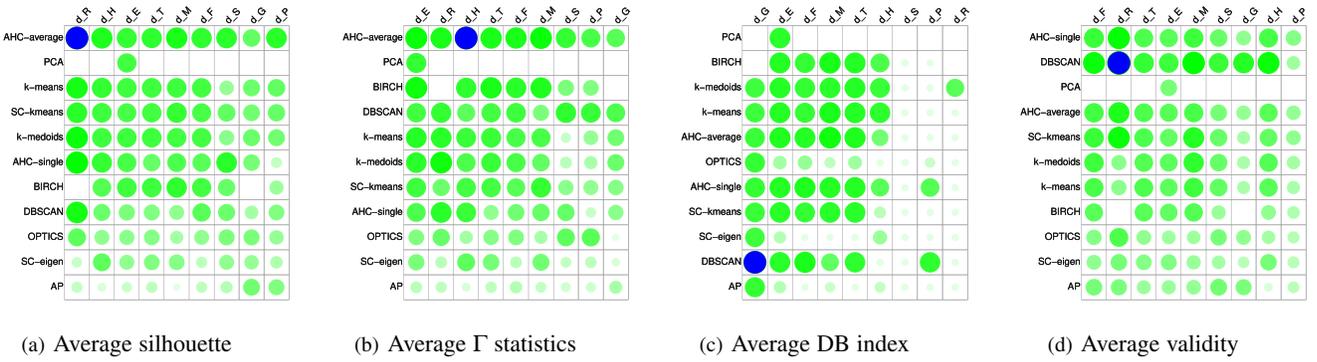


Fig. 4. Ranking visualization of evaluation metrics based on average evaluations of pathline data sets (best cell highlighted in blue). We can see that AHC-average and PCA always result in a better silhouette and Γ statistics, while validity favors AHC-single. Overall, PCA (2.0) and d_E (2.5) respectively have the best evaluation result across all four quality metrics.

on all four evaluations are PCA (1.75), AHC-average (3.25) and k-means (3.75), respectively, and the top three similarity measures are d_R (2.75), d_E (3.75) and d_M (4.0), respectively. Pathlines: We conclude from Fig. 4 that for pathlines the top three clustering techniques are PCA (2.0), AHC-average (2.75) and k-means (4.75), respectively, while the top three similarity measures are d_E (2.5), d_R (3.5) and d_F (3.75) respectively.

Next, we will discuss the ranking results for clustering techniques and similarity measures, respectively, in detail.

4.1.2 Discussion of Clustering Techniques

We note that for both streamlines and pathlines, the top three clustering techniques by ranking-based visualization are PCA, AHC-average and k-means in descending order. As indicated in Sect. 2, these three often generate spherical and convex shapes of clusters, and convex shapes of clusters usually have better evaluation values in silhouette, Γ statistics and DB index. Therefore, the average ranking score of evaluation metrics highlight strong preference towards those clustering techniques that can generate convex shapes of clusters.

The reason for PCA ranking best is that due to PCs (principal components) number determination algorithm designed in [15], i.e., a variance threshold (e.g., 0.99) preset for first r PC components, often results in dimensions lower than 4. Thus either via k-means or AHC-average as post-processing the lower-dimension points can possibly form convex or spherical shapes that lead to favorable scores for silhouette, Γ statistics and DB index. Besides,

compared to many complicated high-dimension similarity measures, *Euclidean distance in lower-dimensional space is easier to achieve rather superior performance similar to point cloud data sets.*

If we omit PCA for being unable to work with other similarity measures and consider other clustering algorithms, we find that AHC-average (ranking order is 3.25) ranks the best while being compatible with all similarity measures introduced in Sect. 3.3. From this perspective we can provide a quantitative reasoning why AHC-average is the most dominant clustering algorithm with various customized similarity measures/metrics in flow visualization as indicated in Sect. 2.2.

In addition, k-means also exhibits relatively good clustering evaluation scores. Compared to PCA and AHC-average, it has lower computational overhead (see performance Table 3 (streamlines) and Table 4 (pathlines) in supplementary document) and memory requirement due to its linearity in both time and memory. This shows great potential of k-means in handling large flow data sets generated from simulation and experiments, as k-means can provide a generally reasonable overview of the flow behavior (suggested by the clustering evaluation metrics) at a much lower cost. However, we should be also aware that the iterative procedure for refining the centers in k-means is strictly deduced from the Euclidean-based similarity measures (i.e., the minimization of distance variation), hence k-means coupled with customized, non-Euclidean similarity measures (e.g., with d_M in [10], [11] and d_G in [72]) has no theoretical foundation despite achieving good clustering results in

flow visualization.

4.1.3 Discussion of Similarity Measures

For streamlines and pathlines, both d_R and d_E are surprisingly good especially for the Γ statistics and silhouette, likely because these metrics favor Euclidean distance based metrics.

d_R is a naive attribute-based similarity measure as described by Oeltze et al. [10], [11], which maps integral curves from original space into one-dimensional space by $\xi : \mathbb{R}^m \rightarrow \mathbb{R}^+$. After this mapping, line clustering is performed on points in \mathbb{R}^+ . Due to unconditional convexity of \mathbb{R}^+ , evaluation metrics like silhouette, Γ and validity measurement tend to exhibit good values. However, the exception is for DB index, $DB = \frac{1}{q} \sum_{i=1}^q \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(c_i, c_j)} \right)$, in which centers of each clusters (c_i, c_j) are still computed in original space, \mathbb{R}^m , instead of \mathbb{R}^+ . After computation, centroid lines are often flattened such that $d(c_i, c_j)$ is often smaller, which generates a quite large DB index value as shown in Fig 3(c) and 4(c).

d_E also ranks high which naturally results from the fact that all the existing clustering techniques and evaluation metrics are established based on the Euclidean distance. We notice a significant difference of the d_E ranking score between streamlines and pathlines, i.e., the ranking score of d_E for streamlines (3.75) is lower than that of pathlines (2.5). This means d_E usually exhibits best evaluation values for the pathline clustering with all clustering techniques, while with streamlines this is not the case. The difference is caused by the fact that the dimension of pathlines is usually low (e.g., 303 for cylinder pathlines) compared to those of streamlines (e.g., usually larger than 1,800), as lower-dimensional Euclidean space, i.e., d_E , often tends to have better DB index of displaying lower intra-cluster distances and high inter-cluster distances. PCA also tends to generate good Γ statistics and DB index which has similarly low dimensions as discussed in Sect. 4.1.2, hence together it can imply d_E is preferable in low-dimensional space rather than high-dimensional space.

However, d_E and d_R are unfortunately not popularly used in the flow visualization community due to the fact that they fail to capture either the spatial proximity or feature related information in the flow. We can consider the similarity measures specifically serving the flow visualization community.

Streamlines: d_M (4.0) is ranked third (better than d_H , d_S , d_P and d_G). Compared to d_R which loses the spatial proximity information of streamlines and d_E which is rarely applied in streamline clustering, d_M encodes more spatial proximity information, therefore, it is quantitatively regarded as the state-of-the-art measure for computing streamline similarity [11] and even for seeding curve searching [85].

Pathlines: d_T (4.0) is ranked fourth (better than d_H , d_M , d_G and d_S and d_P), and it is a revised MCP similarity measure for pathlines considering time overlapping and mismatching. Note that d_T has better evaluation values in clustering evaluation than the conventional d_M , and we quantitatively demonstrate the advantage of d_T over d_M for pathline clustering, which further validates the work by Meuschke et al. [25], [26].

4.2 Visual Inspection

We now discuss the visual comparison of various results obtained using different clustering combinations, which will help determine whether the aforementioned quality metrics are effective or not in indicating the visual quality of the clustering results.

As discussed in Sect. 3.5, different optimal numbers of clusters obtained using either the eigen rotation or the L-methods may fail to provide a fair visual comparison, especially for flow abstraction. This is because different initial numbers of clusters may lead to varying numbers of representative curves for flow abstraction. It is apparent that more representative curves lead to a more informative representation. Hence in Sect. 4.2.2 we use a constant number of clusters as input for all clustering algorithms that require the number of clusters as input. Specifically, we choose the largest number (not larger than 50) from all tested numbers of different similarity measures (see Table 2 in the supplementary document) for each data set, e.g., 30 for Hurricane streamlines. The exception is for blood flow in which we choose four because four cluster representatives are already clear enough to provide a completely visual comparison (see Fig. 12). Meanwhile for those clustering algorithms that do not require a prescribed cluster number, e.g., DBSCAN and OPTICS, our default parameter setting fortunately generates appropriate numbers of clusters for all our testing data sets, which facilitates the subsequent visual comparisons.

From the visual inspection for different visualization tasks, we found that in general, **clustering combinations indicated by quantitative quality metrics do not necessarily provide informative visualization, especially for visual abstraction of streamlines and pathlines.** In the following, we provide a detailed discussion of our findings.

Before providing the detailed discussion, we wish to point out several general observations from the visual inspection of clustering results.

The BIRCH clustering technique always generates thousands of clusters when coupled with d_G and d_R which belong to *shape-based* similarity measures. This implies that BIRCH is better compatible with *spatial proximity based* similarity measures.

AP (Affinity Propagation) clustering (two-level suggested by Tao et al. [38], [39]) sometimes results in either hundreds of single curve clusters or only one cluster for some similarity measures, despite it is better than single-level AP which often generates more than 1000 clusters. This is severely problematic and impractical for visual inspection of the clustering results, since too many or too few clusters either cause occlusion or provide no useful information. Besides, the computational and memory cost for AP is often higher than other methods (note that SC-eigen can reduce computational time by using smaller k). Although AP can combine well with the adapted Procrustes distance [38], [39] and bag-of-features [36], [37], we argue that a deeper investigation should be undertaken to make AP a well-accepted clustering strategy for flow visualization, especially with various customized similarity measures that are specific to a given pattern.

4.2.1 Segmentation

Clustering techniques combined with similarity measures are often used to segment all the streamlines/pathlines into various groups in which candidates inside each group share similar characteristics or properties. Segmentation of integral curves can provide users with more precise and insightful understanding of the flow domain with less redundancy, hence is firstly used for visual inspection of our clustering results.

From the results to be discussed below, we found that **for streamline/pathline segmentation, clustering combinations by best individual quality metrics cannot provide well-separated segmentation compared to those obtained using spatial similarity measures (d_M, d_H).** In contrast, those clustering techniques

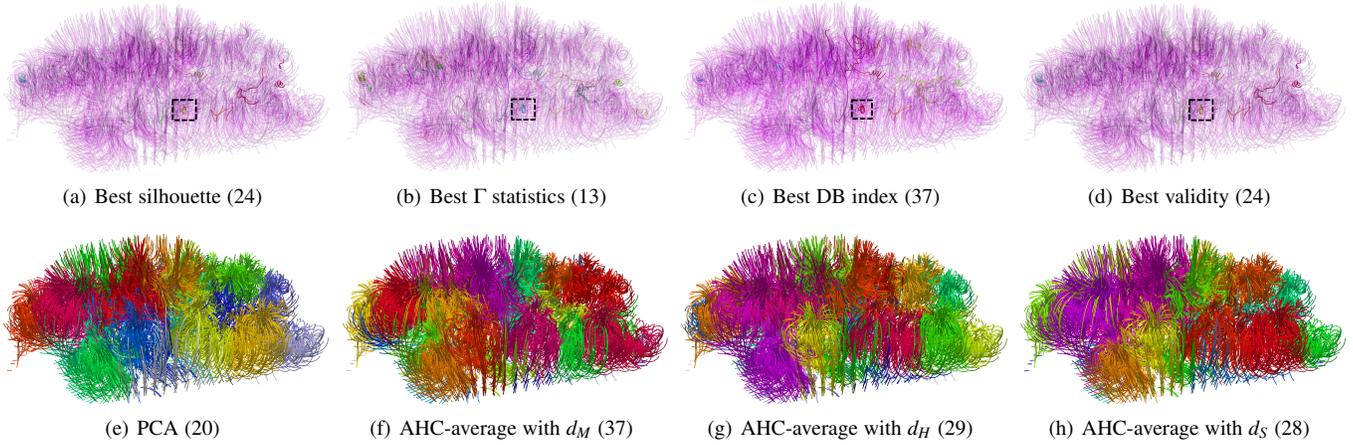


Fig. 5. Crayfish segmentation results of clustering (numbers of clusters are shown in parentheses) according to the best individual evaluation metrics (highlighted as **blue dots** in Fig. 3) (top row) and AHC-average with spatial similarity measures (bottom row). Opacity in (a-d) is adjusted such that individual clusters inside (highlighted in black-dashed rectangle) can be perceived, since inside there exists a cluster containing 99% streamlines due to the chaining effect of the AHC-single. (e)(f) are from average ranking score of streamline quality metrics in Sect. 4.1.2 and (g)(h) are selected spatial segmentations. The top row indicates that the segmentation results cannot capture the spatial proximity and shape similarity of the streamlines despite their respective evaluation values being the best. In contrast, the results shown in the bottom row present a better and clearer segmentation of the streamlines.

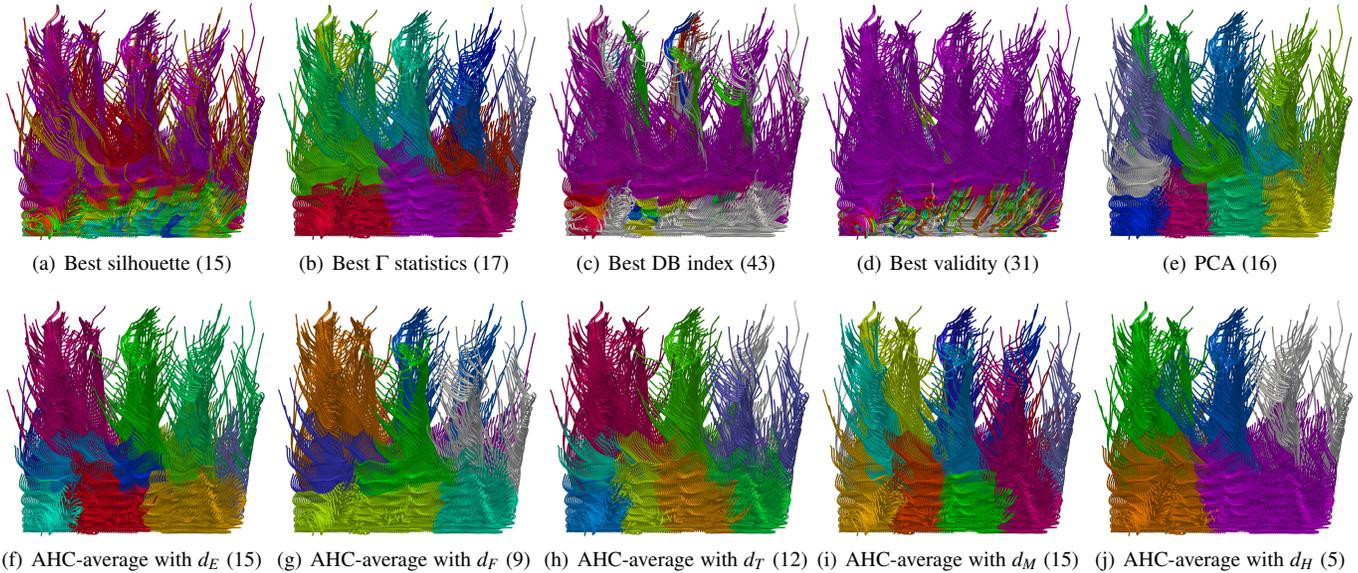


Fig. 6. Tube pathline segmentation results of clustering (numbers of clusters are shown in parentheses) according to the best evaluation metrics highlighted as **blue** in Fig. 4 (see (a-d)), best average ranking-score clustering with highest ranking-score similarity measures (see (e-g)), and visually well-segmented combinations (see (h-j)), respectively.

by average ranking scores (PCA and AHC-average) can. This is because in laminar flow spatially close streamlines/pathlines tend to exhibit similar geometric characteristics.

Streamline Segmentation

For streamlines, we can see from Fig. 3 that AHC-single with d_P has the best silhouette, DB index values and validity measurement on average, while AHC-single with d_R has the best Γ statistics. Therefore, we will visualize the corresponding segmentation results w.r.t. these four clustering combinations for comparison. In addition, PCA and AHC-average are claimed to be on average the top two clustering algorithms based on their ranking scores in Sect. 4.1.3 for streamline data sets. It is also known that clustering with spatial similarity measures, e.g., d_E , d_F , d_M , d_H and d_S , can be used to achieve streamline segmentation. Therefore, we will also choose

the segmentation results obtained using PCA and AHC-average with d_M , d_H and d_S for comparison.

Fig. 5 shows the set of the segmentation results of the crayfish data set selected using the above observation. From this comparison, we see that the results in the bottom row of Fig. 5 provide more reasonable and coherent spatial segmentation for streamlines, even though the top row results have relatively high evaluation metrics. If we correlate the clustering combinations in bottom row with Table 16 of the supplementary document, we observe that the evaluation values of AHC-average with d_M , d_H and d_S are not very different (i.e., the values are in the same order of magnitude) from AHC-single with d_P in silhouette, Γ statistics and DB index, but with more than two-orders of magnitude difference (e.g., $1.2e-4$ compared to $1.2e-6$) in validity.

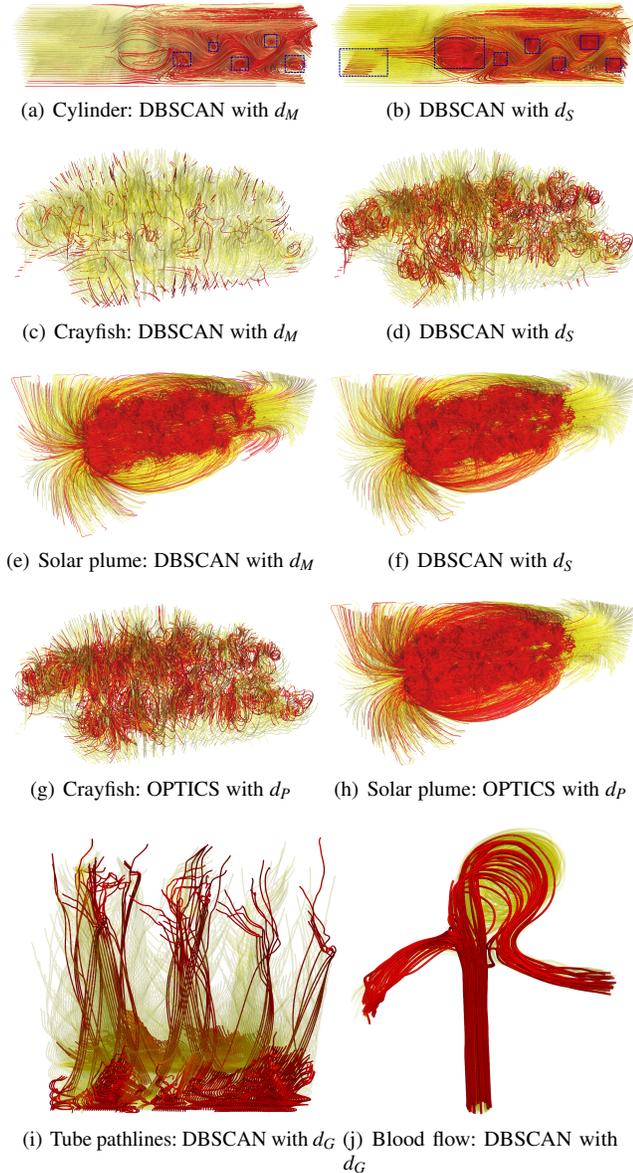


Fig. 7. DBSCAN and OPTICS clustering tends to consider important streamlines/pathlines (e.g., vortex rings or swirling streamlines) as noise with some similarity measures (e.g., d_G , d_S , and d_P). The red streamlines are considered noise, while yellow curves are not. Dashed rectangles in (a) and (b) highlight vortex rings and swirling streamlines that are classified as noise. Only DBSCAN with d_M for the crayfish data suffers less from this (c).

In addition, we observe that AHC-single causes a chaining effect by generating clusters each of which contains only a single curve, as pointed out previously in [10], [11], [15] (see Fig. 5(a-d)), and more than 95% of streamlines are assigned to fewer than two clusters (we call them *dominant clusters*). Therefore, the evaluation metrics in Sect. 3.7 can only be computed on few clusters containing more than two curves and the evaluation values on these *dominant clusters* tend to be quantitatively good despite biased.

Pathline Segmentation

For pathline data sets, we also compare their segmentation results generated with different clustering combinations, according to best individual evaluation metrics (Fig. 6(a-d)), best average ranking-scores in Sect. 4.1.2 (Fig. 6(e-h)), and by manual selection

of best visual segmentation (i.e., AHC-average with d_M (i) and d_H (j)), respectively. In this case, the silhouette favors AHC-average with d_R (a), Γ statistics for AHC-average with d_H (b), DB index for DBSCAN with d_G (c) and validity for DBSCAN with d_R (d). We find that segmentation results selected based on average ranking scores (PCA (e), AHC-average with d_E (f), d_F (g) and d_T (h)) achieve similar quality compared to those obtained using spatial segmentation (e.g., the state-of-the-art AHC-average with d_M (i) and d_H (j)), while segmentation by best individual evaluation metrics cannot.

Limitations of Density-based Clustering

We also discover an intrinsic drawback of density-based clustering segmentation, especially for streamline data sets. That is, the *density-based clustering algorithms inappropriately treat those important or geometrically interesting integral curves as outliers especially with geometry-based similarity measures*, as illustrated in Fig. 7. In particular, combining DBSCAN with geometry-based similarity measures (i.e., d_G , d_R , d_S , d_P) is even worse than with spatial measures (e.g., d_M , d_H). Nonetheless, in either case, some important streamlines (e.g., with strong swirling/rotation configuration) are completely omitted or lost in the output. This can be explained by the fact that DBSCAN considers them as outliers using the respective similarity measures. The reason they are outliers is because those important or geometrical interesting streamlines typically have rather different shapes from the majority of the other streamlines in geometry-based similarity measures, thus, their distance to the majority of the streamlines is very large (possibly much larger than the threshold for the determination of outliers). This in turn causes them to be considered as outliers due to an insufficient number of neighbors. We have observed the similar behavior in the OPTICS results.

Unfortunately, this drawback of the density-based clustering cannot be easily overcome by tuning the parameters, $minPts$ and ϵ . One reason is because trial-and-error parameter tuning is prohibited by the clustering technique itself if the process (especially for streamlines) has a high overhead (see performance of DBSCAN and OPTICS in Table 3 of supplementary document). This important observation indicates that density-based clustering may not be able to generate the desired segmentation results, which is in contrast to the judgment based on the evaluation metrics, in which the validity metric specifically favors DBSCAN (see Sect. 4.1.1). We wish to point out that this observation is consistent with recent work [30] where DBSCAN is applied with dimension-reduced Euclidean distance for feature descriptor of streamlines and stream-surfaces. In contrast to their work focusing on clustering of a low-dimensional space and spatial measures (d_M and d_H) with interactive parameter tuning, we emphasize the drawback of DBSCAN with geometry-based similarity measures (d_G , d_R , d_S and d_P) in original (high-dimensional) space.

However, from Fig. 3(d) and 4(d) we observe that (normalized) validity measurement is biased towards DBSCAN, which contradicts the aforementioned visual observation for DBSCAN. The reason why DBSCAN clustering tends to have the best validity is due to the essential similarity of DBSCAN clustering with validity computation, in which they both use density-based concepts. Given a distance threshold ϵ defined in Sect. 3.2, only pairs whose distances are less than the threshold are considered members of the same cluster by DBSCAN. Therefore, it leads the resulting clusters to exhibit distance homogeneity ($h(\cdot, \cdot)$ in Eq. 2) and density separateness ($g(\cdot, \cdot)$ in Eq. 2) as small as possible computed from the MST (minimal spanning tree) algorithm. Thus, validity

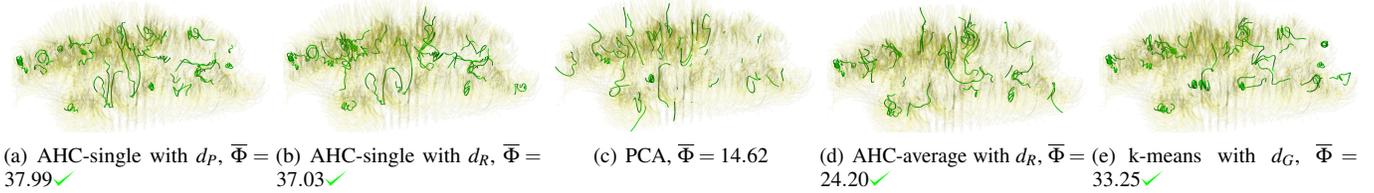


Fig. 8. Visual comparison for the abstraction of crayfish streamlines generated with different clustering algorithms (✓ indicates good abstraction). (a) is obtained with the clustering having the best silhouette and DB index values, (b) has the best Γ statistics value, (c)(d) the abstractions obtained with the clustering having the top average ranking scores, respectively. We also select one abstraction from k-means with d_G in (e). As a comparison, the visually most ideal result is shown in (e) Original streamlines are shown in yellow with lower opacity, while green streamlines are representatives with higher opacity. We find that abstractions indicated by evaluation metrics (except PCA (c)) can effectively select representative streamlines with strong swirling behavior for crayfish streamlines.

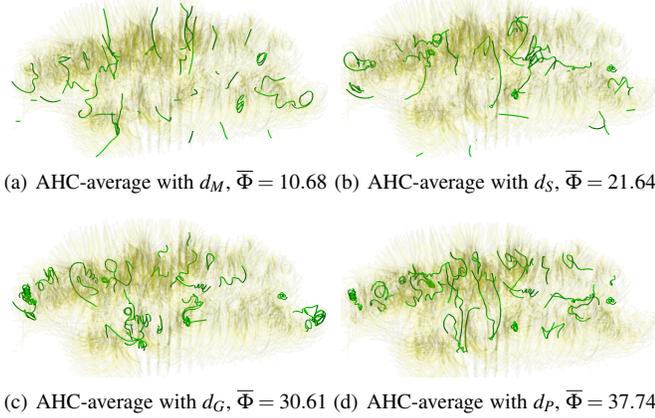


Fig. 9. AHC-average with d_G and d_P can provide better abstraction preserving swirling streamlines than d_M and d_S for crayfish streamlines, even though the latter two exhibit good segmentation results in Fig. 5.

computed for DBSCAN clustering is on average smaller than other clustering algorithms. In this case, quantitative analysis contradicts visual inspection.

4.2.2 Flow Abstraction

After obtaining the segmentation of streamlines/pathlines, reduced representation of the original data can be achieved. The typical strategy is to choose one or more representative curves from each cluster/segment for visualization, as discussed in Sect. 3.6. In this section, we will compare the reduced representations of the tested data sets obtained using different clustering combinations to qualitatively evaluate the clustering quality. The selected representative curves are either the closest or the furthest curve to the centroid of a cluster (Sect. 3.6).

Similarly, we organize our discussion for streamlines and pathline results, separately. For the streamline results, we use the 3D flow behind a cylinder, crayfish, and plume simulation as examples, as they all contain vortical (or rotational/swirling) flows that are interesting to experts. For these simulations, we wish the selected representatives based on the clustering results exhibit as much rotation as possible. To quantify that, we use a metric that measures the total amount of rotation (or directional change) along the individual representative curves [73]. Specifically, we use the average of the total rotation of the individual representative curve, denoted by $\bar{\Phi}$. Intuitively, the larger the value of $\bar{\Phi}$, the more rotation the selected curves exhibit. Note that $\bar{\Phi}$ is not the only determinant for judging the quality of the visual abstraction of

streamlines. In addition to $\bar{\Phi}$, we need to consider which abstraction covers the domain as much as possible if $\bar{\Phi}$ values of several abstractions are similarly high.

For pathlines, we find that from geometric perspective, physically interesting pathlines should also exhibit high rotations which means they are spiralling or circulating in this spatio-temporal domain. Hence, we also recommend observing judge which pathline abstraction captures more such important features based on the aforementioned $\bar{\Phi}$, and spatial coverage as well, e.g., vortex regions and two subregions highlighted in blue-dashed area in Fig. 12(j).

In general, **considering visual abstraction with aforementioned objectives, clustering combinations identified by the best individual quality metrics fail to present desired abstraction, while those by average ranking scores may sometimes, but not always, lead to desired abstractions.**

Streamline abstraction

Fig. 8 (a-d) shows the reduced representations of the crayfish data that are generated with the clustering combinations identified by individual ((a) and (b)) and average quality metrics ((c) and (d)) with the best quality, respectively. As a comparison, a representation that visually reveals more vortical behaviors (or vortices) in this data is shown in (e). In these visualizations, the green curves are the representative streamlines, while the yellow transparent curves are the original input streamlines. We see that clustering combinations with the top individual quality metric values and top average ranking scores (except PCA) can select streamlines with the desired swirling behavior as well as in Fig. 8(e). Among the four representations suggested by the quality metrics, AHC-single with d_P seems to perform the best both visually and quantitatively (i.e., having large $\bar{\Phi}$ value). This is in fact due to the chaining effect that tends to separate the streamlines with swirling configurations into individual clusters.

We also compare d_M and d_H to d_P and d_G when used with AHC-average clustering since the latter two both perform well in the average ranking score of evaluation and the visual inspection in streamline segmentation for the crayfish data (see Fig. 5). We find that in general d_P and d_G demonstrate a better ability to preserve the swirling streamlines in a reduced representation of the crayfish streamlines.

Fig. 10 shows the reduced representations for the plume (top row) and the 3D flow behind cylinder (bottom row) data sets. For each row, we again show the representation suggested by the quality metrics and visual inspection, respectively. From this comparison, we observe that AHC-average with d_M (see Fig. 10(b), 10(g)) and d_H (see Fig. 10(d)) and PCA (see Fig. 10(c)) are often

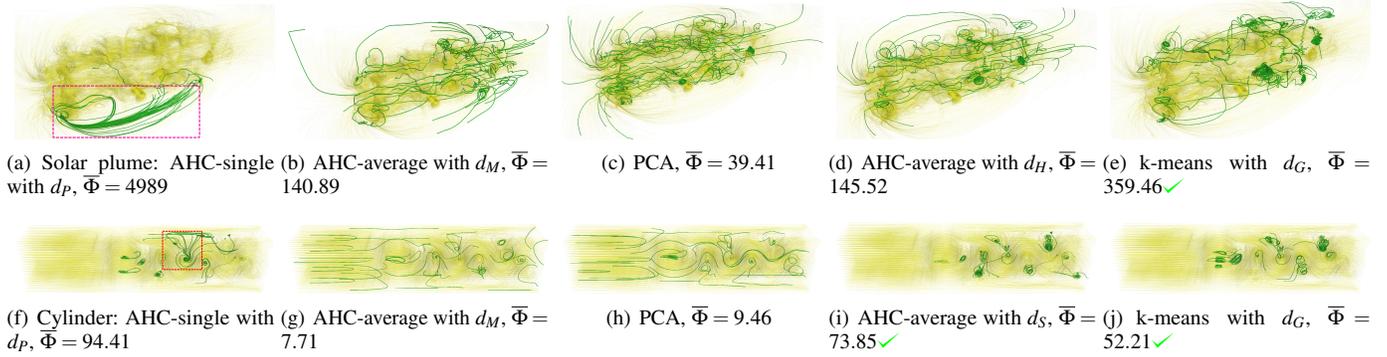


Fig. 10. Visual abstraction of streamlines from solar plume (top row) and cylinder flow (bottom row) by clustering algorithms (✓ marks better abstraction). The yellow are original streamlines with lower opacity while green streamlines are representatives with higher opacity. We find that PCA and clustering algorithms with d_M and d_H characterize the overview of the flow, while clustering with d_p , d_S and d_G tends to highlight and preserve swirling streamlines. Additionally, d_G and d_S are more stable than d_p because d_p is sometimes trapped in erratic boundaries instead of vortex rings due to a chaining effect, see outlying boundary streamlines in dashed rectangle of (a)(f).

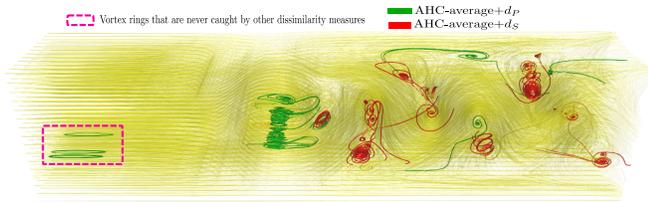


Fig. 11. Abstraction combination of AHC-average with d_p (green) and d_S (red) for cylinder streamlines. Each of them may capture partial features inside the domain. For example, d_p extracts lower left vortices (highlighted in the dashed red rectangle) that cannot be captured with d_S . Combining them together can create a much better and more thorough abstract visualization.

effective in conveying the overview and overall structures of the flow, while AHC-average/single with d_G , d_R , d_S , and d_p , and k-means with d_G tend to highlight and characterize the vortex details. Meanwhile, d_p is not stable in feature highlighting. d_p suffers from a chaining effect not only in AHC-single but also other clustering algorithms (e.g., AHC-average), which results in preferences of large bundles of boundary streamlines (see red dashed part in Fig. 10(a) and 10(f)), instead of internal swirling streamlines. Sometimes it is able to extract vortex rings that no other measures can (see red dashed area in Fig. 11).

From the three reduced streamline representations, we conclude that clustering algorithms with d_G , d_R and d_S are more robust and stable for feature highlighting in flow abstraction than d_p . k-means with d_G [72] and d_R are able to capture streamlines with large curvature variation with similar quality to AHC-average with d_S [13], while having the lowest computation cost (see Table 3 in the supplementary document). Individual quality metrics (e.g., AHC-single with d_p) cannot provide a visually desired abstraction, while an average ranking score of quality metrics can partially (e.g., AHC-average with d_R) achieve this objective.

Combining multiple abstractions In most cases, abstraction with one single similarity measure can only capture a subset of features reflected by the streamlines, and it can be helpful to combine multiple abstractions to make a more complete visual representation of the important features. For example, in Fig. 11 AHC-average with d_p can capture a sequence of near closed streamlines (highlighted in the dashed red rectangle) while failing to highlight a number of other swirling streamlines. In contrast, AHC-

average with d_S can detect more vortical features than d_p but still misses the bottom-left vortex rings. Combining the representations obtained from the two clustering results leads to a more complete visualization of the flow.

This conclusion is also observable for pathline abstraction, e.g., in blood flow abstraction results (see Fig. 12), AHC-average with d_H (h) combined with PCA (i) can generate the required abstraction that preserves each highlighted feature similar to AHC-average with d_M (k). However, we think it is unnecessary to mention the benefit of combinations in pathline abstraction since a single abstraction (like (k)) already generates good abstraction results. This is in contrast to streamline abstraction where no single abstraction can capture the complete and complex features.

Pathline abstraction

Similarly, we compare the reduced representation suggested by the quality metrics. Guided by our experience that density-based clustering tends to classify streamlines/pathlines with distinct swirling behavior as outliers (see Fig. 7 and Sect. 4.2.1), hence we only consider results obtained using AHC-average with d_R (by best silhouette in Fig. 4(a)) and AHC-average with d_H (by best Γ statistics in Fig. 4(b)). In addition, the abstractions suggested by the average ranking-score (i.e., AHC-average with d_T and PCA) are also selected. Further, we select one or two reduced representations, that can preserve pathlines of important features, e.g., pathlines around the vortices behind cylinder in the cylinder pathlines (see blue-dashed area in Fig. 12(e)), and vortex regions and two subregions in the blood flow (see blue-dashed areas in Fig. 12(j)), as an additional visual comparison for the aforementioned abstraction from quantitative analysis.

From Fig. 12 we can see that abstractions suggested by individual evaluation metrics (i.e., AHC-average with d_H in Fig. 12(b)(h)) and average ranking scores (i.e., PCA in Fig. 12(c)(i)) lose more or less the aforementioned pathline features. However, in general AHC-average with shape-based similarity measures, i.e., d_S (see Fig. 15 in the supplementary document), d_R and d_p , and k-means with d_G and d_R are able to capture these important pathlines in the abstraction in the given pathline data sets. Specifically, for cylinder pathlines, AHC-average with d_p (Fig. 12(e)) and k-means with d_R (Fig. 12(f)) provide better focus on the vortical features than AHC-average with d_R (Fig. 12(a)). While for blood flow, AHC-average with d_R (Fig. 12(g)) and d_M (Fig. 12(k)) generate the best abstraction results by preserving not only the outlier vortices

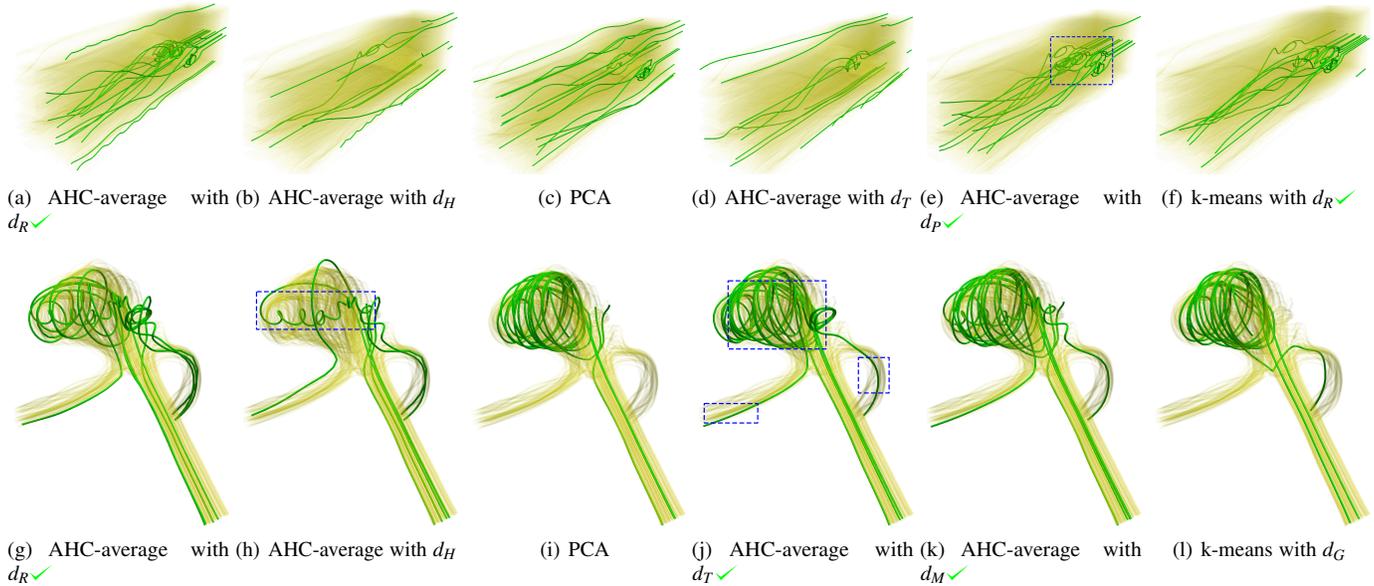


Fig. 12. Cylinder pathline (top row) and blood flow (bottom row) abstraction (✓ indicates good abstraction). The highlighted areas in those blue-dash rectangles of cylinder pathlines (e) and blood flow (j) are pathline features that abstraction should capture. Abstraction by individual evaluation metrics ((b)(h)) and top average ranking scores (e.g., PCA in (c)(i) and AHC-average with d_T in (d)) only provide partial information for the pathlines inside the domain. Generally, AHC-average with *shape-based* similarity measures (d_R , d_S , d_P) and k-means with d_R can roughly capture the full features of given pathline data sets.

but also the vortex cores (see blue-dashed area in Fig. 12(h)) in the center region of the blood flow.

We conclude that *AHC-average with shape-based similarity measures (d_R , d_S and d_P) and k-means with d_R (k-means with d_G is not good for cylinder pathline abstraction in Fig. 15 of supplementary document) can generate better representation of pathlines than those indicated by best individuals or averaged ranking scores of evaluation metrics.*

4.3 Empirical Guidelines

By comparing the quantitative analysis and visual inspection of the clustering quality for streamlines/pathlines, we see that they need not agree with each other. This in part explains that most of the clustering approaches for geometric-based flow visualization do not rely on the well-established quantitative metrics to identify the most effective clustering technique and distance measures for their specific flow data. On the other hand, purely relying on visual inspection to determine the clustering result quality can be subjective and affected by many visualization factors (e.g., the rendering of the lines, view points, lighting, etc., as described in [86], [87], [88]). Nonetheless, based on the above assessment of the clustering results, we can offer the following guidance for streamline/pathline clustering in flow visualization.

1) For streamlines, PCA and AHC-average are the top two clustering algorithms, and d_R , d_E and d_M are the best similarity measures, based on the quantitative evaluation ranking-score (see Sect. 4.1.1). For pathlines, PCA and AHC-average are the top three clustering algorithms, and d_E and d_R are the best similarity measures (see Sect. 4.1.1). PCA clustering works well in the above quantitative evaluation analysis due to its applicability in Euclidean distance of low-dimensional space. In contrast, AHC-average suffers less from a chaining effect and is widely applied due to its robustness and versatile with various, customized similarity metrics.

- 2) We quantitatively validate the reason why d_M for streamlines and d_T for pathlines are the state-of-the-art similarity measures in integral curve based clustering (see Sect. 4.1.3).
- 3) Density-based clustering algorithms, despite being strongly favored by validity measurement, tend to treat important (or geometrically interesting) streamlines/pathlines as outliers, thus, are not recommended for flow visualization, especially with *shape-based* similarity measures (see Sect. 4.2.1).
- 4) AP clustering (including two-level) often generates too many or few clusters at a higher computational cost. BIRCH is not compatible with *shape-based* similarity measures.
- 5) For the task of segmenting streamlines/pathlines, clustering algorithms with top average ranking scores (e.g., PCA and AHC-average with d_M for streamlines, PCA and AHC-average with d_T and d_E for pathlines) can provide a better spatial segmentation than those obtained using the clustering combinations with the highest individual evaluation metrics. Besides, visual inspection also suggests that AHC-average with d_M , d_H , d_S and d_T can generate the most ideal segmentation of the streamline/pathline data sets. PCA produces good spatial segmentation for both streamlines/pathlines, which also exhibits on average good evaluation scores. Generally, a segmentation is best achieved by clustering algorithms with spatial similarity measures.
- 6) To generate a reduced representation for sets of densely placed streamlines, clustering algorithms (especially AHC-average) with the similarity measures d_R , d_G and d_S are more robust than with d_P in highlighting vortex structure and swirling streamlines, and k-means with d_G and d_R can also produce a desired reduced representation but at a much lower cost. For pathline abstraction, AHC-average with *shape-based* similarity measures (d_R , d_S and d_P) and k-means with d_R are more preferred in preserving important features of pathlines. Unfortunately, these visual inspection results of abstraction

often do not match with the results obtained via quantitative evaluation.

- 7) d_S [13] in general works well in both segmentation and reduced representation for streamlines because d_S considers both spatial and shape similarity of streamlines. This indicates that similarity measures that linearly combine spatial and signature measures can be potentially helpful to flow visualization.
- 8) Compared to setting the input number of clusters for some clustering algorithms (i.e., k-means, k-medoids, AHC, SC k-means, and PCA), setting the values of ϵ (for DBSCAN and OPTICS) or distance threshold (for BIRCH) or preference initialization (AP) is usually more difficult and requires prior knowledge of the data sets. Hence in general, the clustering algorithms that only need to set the number of clusters are more popular (especially AHC-average) for processing data sets without prior knowledge. This observation is similar to the discussion between AHC and DBSCAN by Meuschke et al. [25], [26].
- 9) Low-dimensional Euclidean distance (d_E) demonstrates good overall evaluation results in clustering analysis (see Sect. 4.1.1). Besides, d_E in lower-dimensional space features advantages of numerical efficiency, theoretical stability and predictable shapes of clusters. These important advantages lead to burgeoning work in flow visualization to perform clustering after dimension-reduction techniques for streamlines/pathlines, e.g., PCA in streamline variability plot [15] and t-SNE for feature descriptors after auto-encoder learning [30]. We believe after proper dimension reduction, d_E combined with a carefully chosen clustering algorithm will form a trend in clustering frameworks for flow visualization.
- 10) Attribute-based streamline distances (e.g., d_R in our experiment, linear and angular entropy in [18], and streamline attributes in [10], [11]) usually exhibit good evaluation scores for clustering analysis due to their ability in mapping high-dimensional streamlines to lower-dimensional Euclidean spaces. That said, the design of similarity measures that aim to emphasize given flow characteristics can focus on the relevant attribute of integral curves, which can be beneficial in performing clustering in a low-dimensional Euclidean space.
- 11) Sometimes combining the reduced representations generated with two or more appropriate clustering techniques may capture more complete sets of features in streamlines as illustrated in Fig. 11.

Finally, choosing appropriate clustering algorithms and similarity measures is a complex problem, which needs to consider not only the aforementioned visual/quantitative guidance, but also other aspects of the data, e.g., the size of data sets and the required computation time, whether focusing on spatial or shape difference, or customized similarity measures for specific purposes. In most cases, similarity measures designed in flow visualization literature are not rigorous mathematical metrics, therefore, it is not possible to theoretically derive the property of metric space, like its topology discussed in [35] or convex analysis in [60], nor can shapes of clusters be predicted so that a proper clustering can be determined and applied. There exists inextricable disparity between theory and application for clustering in flow visualization, and the qualitative (visual) and quantitative (analytic) are often not compatible for clustering results, which still requires in-depth investigation.

4.4 Comparison with Blood Flow Analysis Work

To our best knowledge, the only existing work on clustering analysis (visually and quantitatively) for flow visualization is the visualization of blood flow [10], [11]. Despite this work shares some similarity to that work, e.g., both use silhouette and Γ statistics for evaluation, and both apply tube-based rendering for visualization, there are important differences between these two.

- We evaluate a much larger set of clustering algorithms and similarity measures, while blood flow analysis work only considers d_M with k-means, SC with eigen-rotation and four types of AHC. We experiment on both streamline and pathline data sets with different flow characteristics, compared to blood flow data sets. That said, our conclusions are more general.
- We adopt a novel normalized validity measurement which proves to be more effective and general in evaluating clustering quality for point-based data sets than silhouette and Γ statistics (both of which only work for convex clusters of shape), and further investigate the drawback of only using validity as quantitative reference in flow visualization.
- We employ detailed quantitative analysis and visual comparisons based on either individuals or average ranking score of the evaluation metrics, and thoroughly discuss the disparity between quantitative and visual preferences for different clustering combinations.
- Our quantitative conclusion for AHC-average in streamlines (see Sect. 4.1) is consistent with blood flow where both by silhouette and Γ statistics AHC-average works well. However, in the blood flow visualization, SC with eigen-rotation was claimed to exhibit good evaluation based on silhouette and Γ statistics (see Fig. 11 in the supplementary document where SC-eigen with d_M exhibits the best three evaluation scores), while in our evaluation it is actually one of the worst of all clustering algorithms. We argue that the disparity for ranking of SC-eigen w.r.t. silhouette and Γ statistics is that we adopt an average evaluation over different similarity measures other than just d_M , and quantitative conclusions from blood flow simulation cannot be simply extended into other streamline data sets.
- The conclusion made by [10], [11] that SC (SC-eigen in our paper) is the best clustering technique only applies to the blood flow data set based on our experimental results. We believe it is due to the simpler and clearer flow features of the blood flow (see blood flow abstraction in Sect. 4.2.2) when compared to other physically sophisticated flows, e.g., crayfish and plume, with more hidden vortices. Besides, we found no further applications of SC to other flow data sets except blood flow after the streamline embedding work [35] (TVCG 2012). This is likely because SC (especially SC-eigen) has a high overhead in determining the optimal number of clusters for simulated flow data with a large value of k . Compared to blood flow whose k is not larger than 20 [10], [11], we usually do not have a-priori knowledge of k for general simulated flow data, hence, we have to set k to a large number which results in a longer computation time. Therefore, SC is often impractical for interactive exploration of large-scale flow data sets, while AHC is preferred instead (see details in Sect. 9 of the supplementary document).

Note that, Han et al. [30] recently compared different clustering algorithms for feature descriptors of streamlines and stream surfaces through different dimensional reduction techniques. However,

their work is purely qualitative and focusing on feature descriptors after autoencoder learning, while our work is both quantitatively and qualitatively in original integral line space.

5 CONCLUSION AND FUTURE WORK

In this paper we perform a comprehensive evaluation of popular clustering techniques coupled with a number of popular similarity measures. We use both quantitative analysis and visual inspection to assess the clustering results, and derive empirical guidance (see Sect. 4.3) for selecting an appropriate clustering and similarity measure combination. This is the first work, to the best of our knowledge, that attempts a comprehensive experimental study on a large number of clustering techniques and similarity measures for integral curve clustering. We believe the outcome of this study will benefit the practice of flow data visualization in achieving a reduced representation for streamline/pathline data sets.

Limitations and future work There are a number of limitations of our current study that we wish to improve in future work.

First, we use a distance-based representative approach for visual inspection of clustering results, and it may be prone to specific patterns, e.g., vortical structures in the streamlines. In the future we want to investigate several existing representative approaches to find the optimal method that can characterize the clusters of integral curves. In addition, there are more features or patterns conveyed in the streamlines (e.g., separation lines), and we wish to further explore them via clustering techniques.

Second, the similarity measures (except for d_T) are generally suitable for streamlines, and we would in the future investigate those specific similarity measures for pathlines to derive a more accurate guidance on pathline clustering. Also visual reference for judging pathline abstraction might be not accurate and precise, and we would like to extend our experimental work to more complex and meaningful pathline data sets.

Third, clustering methods based on statistical analysis and information are not frequently applied in flow visualization, and they are mostly viewed as inferior for comparison. For example, PCA+GMM (Gaussian mixture model) is compared to PCA+AHC, and the latter is shown to be better for streamline variability clustering [15]. We would like to extend these statistical-based methods to integral curve clustering in the future, either in representative selection under functional decomposition [78], or using model-based clustering for turbulent combustion particle data [89]. Due to solid theory and well-constructed objectives of statistical-based methods, we believe they can exhibit enormous potential and versatile applications in conveying effective abstraction of complicated flow data sets.

ACKNOWLEDGMENTS

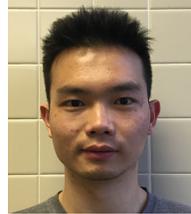
This work is supported by National Science Foundation (IIS 1553329). We would like to thank the anonymous reviewers of IEEE VIS 2018 and TVCG for providing instructive comments to our work. We also wish to thank Duong Binh Nguyen for providing the streamline and pathline data sets for the evaluation.

REFERENCES

- [1] Z. Liu, R. Moorhead, and J. Groner, "An advanced evenly-spaced streamline placement algorithm," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 965–972, Sep. 2006. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2006.116>
- [2] K. Wu, Z. Liu, S. Zhang, and R. J. M. II, "Topology-aware evenly spaced streamline placement," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 5, pp. 791–801, Sept 2010.
- [3] W. Zhang, B. Sun, and Y. Wang, "A streamline placement method highlighting flow field topology," in *2010 International Conference on Computational Intelligence and Security*, Dec 2010, pp. 238–242.
- [4] S. Furuya and T. Itoh, "A streamline selection technique for integrated scalar and vector visualization," 2008.
- [5] L. Xu, T. Y. Lee, and H. W. Shen, "An information-theoretic framework for flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 6, pp. 1216–1224, Nov 2010.
- [6] C. Wang and H.-W. Shen, "Information theory in scientific visualization," *Entropy*, vol. 13, pp. 254–273, 2011.
- [7] T. Y. Lee, O. Mishchenko, H. W. Shen, and R. Crawfis, "View point evaluation and streamline filtering for flow visualization," in *2011 IEEE Pacific Visualization Symposium*, March 2011, pp. 83–90.
- [8] J. Tao, J. Ma, C. Wang, and C. K. Shene, "A unified approach to streamline selection and viewpoint selection for 3d flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 3, pp. 393–406, March 2013.
- [9] C.-K. S. Jun Ma, Chaoli Wang, "Coherent view-dependent streamline selection for importance-driven flow visualization," vol. 8654, 2013, pp. 8654 – 8654 – 15. [Online]. Available: <https://doi.org/10.1117/12.2001887>
- [10] S. Oeltze, D. J. Lehmann, H. Theisel, and B. Preim, "Evaluation of streamline clustering techniques for blood flow data," 2012.
- [11] S. Oeltze, D. J. Lehmann, A. Kuhn, G. Janiga, H. Theisel, and B. Preim, "Blood flow clustering and applications in virtual stenting of intracranial aneurysms," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 5, pp. 686–701, May 2014.
- [12] R. Jianu, C. Demiralp, and D. Laidlaw, "Exploring 3d dti fiber tracts with linked 2d representations," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1449–1456, Nov. 2009. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2009.141>
- [13] T. McLoughlin, M. W. Jones, R. S. Laramee, R. Malki, I. Masters, and C. D. Hansen, "Similarity measures for enhancing interactive streamline seeding," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 8, pp. 1342–1353, Aug 2013.
- [14] J. Wei, C. Wang, H. Yu, and K. L. Ma, "A sketch-based interface for classifying and visualizing vector fields," in *2010 IEEE Pacific Visualization Symposium (PacificVis)*, March 2010, pp. 129–136.
- [15] F. Ferstl, K. Brger, and R. Westermann, "Streamline variability plots for characterizing the uncertainty in vector field ensembles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 1, pp. 767–776, Jan 2016.
- [16] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, no. 2, pp. 165–193, Jun 2015. [Online]. Available: <https://doi.org/10.1007/s40745-015-0040-1>
- [17] H. Yu, C. Wang, C.-K. Shene, and J. H. Chen, "Hierarchical streamline bundles," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 8, pp. 1353–1367, Aug. 2012. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2011.155>
- [18] C.-K. Chen, S. Yan, H. Yu, N. Max, and K.-L. Ma, "An illustrative visualization framework for 3d vector fields," *Comput. Graph. Forum*, vol. 30, no. 7, pp. 1941–1951, 2011. [Online]. Available: <http://dblp.uni-trier.de/db/journals/cgf/cgf30.html/ChenYYMM11>
- [19] S. Oeltze-Jafra, J. R. Cebra, G. Janiga, and B. Preim, "Cluster analysis of vortical flow in simulations of cerebral aneurysm hemodynamics," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, pp. 757–766, 2016.
- [20] G. G. I. Corouge and S. Gouttard, "Towards a shape model of white matter fiber bundles using diffusion tensor mri," in *2004 IEEE International Symposium on Biomedical Imaging (ISBI)*, April 2004, pp. 344–347.
- [21] B. Moberts, A. Vilanova, and J. J. van Wijk, "Evaluation of fiber clustering methods for diffusion tensor imaging," in *VIS 05. IEEE Visualization, 2005.*, Oct 2005, pp. 65–72.
- [22] W. Chen, S. Zhang, S. Correia, and D. S. Ebert, "Abstractive representation and exploration of hierarchically clustered diffusion tensor fiber tracts," *Computer Graphics Forum*, vol. 27, no. 3, pp. 1071–1078, 2008. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2008.01244.x>
- [23] P. Klemm, K. Lawonn, M. Rak, B. Preim, K. Toennies, K. Hegenscheid, H. Vizke, and S. Oeltze, "Visualization and Analysis of Lumbar Spine Canal Variability in Cohort Study Data," in *Vision, Modeling & Visualization*, M. Bronstein, J. Favre, and K. Hormann, Eds. The Eurographics Association, 2013.
- [24] S. Zhang, S. Correia, and D. H. Laidlaw, "Identifying white-matter fiber bundles in dti data using an automated proximity-based fiber-clustering

- method," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 5, pp. 1044–1053, Sept 2008.
- [25] M. Meuschke, S. Voß, B. Preim, and K. Lawonn, "Exploration of blood flow patterns in cerebral aneurysms during the cardiac cycle," *Computers & Graphics*, vol. 72, pp. 12–25, 2018. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0097849318300128>
- [26] M. Meuschke, S. Oeltze-Jafra, O. Beuing, B. Preim, and K. Lawonn, "Classification of blood flow patterns in cerebral aneurysms," *IEEE Transactions on Visualization and Computer Graphics*, pp. 1–1, 2018.
- [27] Z. Wang, J. M. Esturo, H.-P. Seidel, and T. Weinkauff, "Stream linebased pattern search in flows," *Computer Graphics Forum*, pp. n/a–n/a, 2016. [Online]. Available: <http://dx.doi.org/10.1111/cgf.12990>
- [28] K. Lu, A. Chaudhuri, T. Y. Lee, H. W. Shen, and P. C. Wong, "Exploring vector fields with distribution-based streamline analysis," in *2013 IEEE Pacific Visualization Symposium (PacificVis)*, Feb 2013, pp. 257–264.
- [29] J. Ma, C. Wang, and C. K. Shene, "Flowgraph: A compound hierarchical graph for flow field exploration," in *2013 IEEE Pacific Visualization Symposium (PacificVis)*, Feb 2013, pp. 233–240.
- [30] J. Han, J. Tao, and C. Wang, "Flownet: A deep learning framework for clustering and selection of streamlines and stream surfaces," *IEEE Transactions on Visualization and Computer Graphics*, 2019.
- [31] M. Meuschke, K. Lawonn, B. Köhler, U. Preim, and B. Preim, "Clustering of aortic vortex flow in cardiac 4d pc-mri data," in *Bildverarbeitung für die Medizin 2016*, T. Tolxdorff, T. M. Deserno, H. Handels, and H.-P. Meinzer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 182–187.
- [32] R. Englund, T. Ropinski, and I. Hotz, "Coherence maps for blood flow exploration," in *Proceedings of the Eurographics Workshop on Visual Computing for Biology and Medicine*, ser. VCBM '16. Goslar Germany, Germany: Eurographics Association, 2016, pp. 79–88. [Online]. Available: <https://doi.org/10.2312/vcbm.20161274>
- [33] L. O'Donnell, M. Kubicki, M. E. Shenton, M. H. Dreusicke, W. E. L. Grimson, and C.-F. Westin, "A method for clustering white matter fiber tracts," *American Journal of Neuroradiology*, vol. 27, p. 1032–1036, 2006.
- [34] T. Schultz and G. L. Kindlmann, "Open-box spectral clustering: Applications to medical image analysis," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2100–2108, Dec 2013.
- [35] C. Rossl and H. Theisel, "Streamline embedding for 3d vector field exploration," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 3, pp. 407–420, March 2012.
- [36] C.-K. S. Yifei Li, Chaoli Wang, "Streamline similarity analysis using bag-of-features," pp. 9017 – 9017 – 12, 2014. [Online]. Available: <http://dx.doi.org/10.1117/12.2038253>
- [37] Y. Li, C. Wang, and C.-K. Shene, "Extracting flow features via supervised streamline segmentation," *Comput. Graph.*, vol. 52, no. C, pp. 79–92, Nov. 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.cag.2015.06.003>
- [38] J. Tao, C. Wang, and C. K. Shene, "Flowstring: Partial streamline matching using shape invariant similarity measure for exploratory flow visualization," in *2014 IEEE Pacific Visualization Symposium*, March 2014, pp. 9–16.
- [39] J. Tao, C. Wang, C. K. Shene, and R. A. Shaw, "A vocabulary approach to partial streamline matching and exploratory flow visualization," *IEEE Transactions on Visualization and Computer Graphics*, vol. 22, no. 5, pp. 1503–1516, May 2016.
- [40] P.-N. Tan, M. Steinbach, and V. Kumar, "Introduction to data mining. (first edition)." Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2005.
- [41] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, May 2005.
- [42] L. Kaufman and P. Rousseeuw, *Clustering by Means of Medoids*, ser. Delft University of Technology : reports of the Faculty of Technical Mathematics and Informatics. Faculty of Mathematics and Informatics, 1987. [Online]. Available: <https://books.google.com/books?id=HK-4GwAACAAJ>
- [43] J. Blömer, C. Lammersen, M. Schmidt, and C. Sohler, *Theoretical Analysis of the k-Means Algorithm – A Survey*. Cham: Springer International Publishing, 2016, pp. 81–116. [Online]. Available: https://doi.org/10.1007/978-3-319-49487-6_3
- [44] A. M. Mehar, K. Matawie, and A. Maeder, "Determining an optimal value of k in k-means clustering," in *2013 IEEE International Conference on Bioinformatics and Biomedicine*, Dec 2013, pp. 51–55.
- [45] D. J. Ketchen and C. L. Shook, "The application of cluster analysis in strategic management research: An analysis and critique," *Strategic Management Journal*, vol. 17, no. 6, pp. 441–458, 1996. [Online]. Available: [http://dx.doi.org/10.1002/\(SICI\)1097-0266\(199606\)17:6<441::AID-SMJ819>3.0.CO;2-G](http://dx.doi.org/10.1002/(SICI)1097-0266(199606)17:6<441::AID-SMJ819>3.0.CO;2-G)
- [46] D. Arthur and S. Vassilvitskii, "K-means++: The advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA '07. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1283383.1283494>
- [47] D. Napoleon, M. Praneesh, S. Sathya, and M. SivaSubramani, "An efficient numerical method for the prediction of clusters using k-means clustering algorithm with bisection method," in *Global Trends in Information Systems and Software Applications*, P. V. Krishna, M. R. Babu, and E. Ariwa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 256–266.
- [48] L. Rokach and O. Maimon, *Clustering Methods*. Boston, MA: Springer US, 2005, pp. 321–352. [Online]. Available: https://doi.org/10.1007/0-387-25465-X_15
- [49] T. Zhang, R. R., and M. Livny, "Birch: A new data clustering algorithm and its applications," *Data Mining and Knowledge Discovery*, vol. 1, no. 2, pp. 141–182, 1997.
- [50] V. Cohen-Addad, V. Kanade, F. Mallmann-Trenn, and C. Mathieu, "Hierarchical clustering: Objective functions and algorithms," in *SODA*, 2018.
- [51] H.-P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 3, pp. 231–240. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.30>
- [52] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, ser. KDD'96. AAAI Press, 1996, pp. 226–231. [Online]. Available: <http://dl.acm.org/citation.cfm?id=3001460.3001507>
- [53] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: A partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '07. New York, NY, USA: ACM, 2007, pp. 593–604. [Online]. Available: <http://doi.acm.org/10.1145/1247480.1247546>
- [54] M. Ankerst, M. M. Breunig, H. Peter Kriegel, and J. Sander, "Optics: Ordering points to identify the clustering structure." ACM Press, 1999, pp. 49–60.
- [55] X. Hu, J. Huang, M. Qiu, C. Chen, and W. Chu, "Ps-dbscan: An efficient parallel dbscan algorithm based on platform of ai (pai)," *CoRR*, vol. abs/1711.01034, 2017.
- [56] X. Hu, J. Huang, and M. Qiu, "A communication efficient parallel dbscan algorithm based on parameter server," in *Proceedings of the 2017 ACM Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: ACM, 2017, pp. 2107–2110. [Online]. Available: <http://doi.acm.org/10.1145/3132847.3133112>
- [57] F. Huang, Q. Zhu, J. Zhou, J. Tao, X. Zhou, D. Jin, X. Tan, and L. Wang, "Research on the parallelization of the dbscan clustering algorithm for spatial data mining based on the spark platform," *Remote Sensing*, vol. 9, no. 12, 2017.
- [58] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000. [Online]. Available: <http://dx.doi.org/10.1109/34.868688>
- [59] U. Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, Dec. 2007. [Online]. Available: <http://dx.doi.org/10.1007/s11222-007-9033-z>
- [60] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS*. MIT Press, 2001, pp. 849–856.
- [61] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proceedings of the 17th International Conference on Neural Information Processing Systems*, ser. NIPS'04. Cambridge, MA, USA: MIT Press, 2004, pp. 1601–1608. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2976040.2976241>
- [62] J. Demmel, I. Dumitriu, and O. Holtz, "Fast linear algebra is stable," *Numerische Mathematik*, vol. 108, no. 1, pp. 59–91, Nov 2007. [Online]. Available: <https://doi.org/10.1007/s00211-007-0114-x>
- [63] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007. [Online]. Available: <http://science.sciencemag.org/content/315/5814/972>
- [64] W. Arter and A. C. Newell, "Numerical simulation of rayleighbernard convection in shallow tanks," *The Physics of Fluids*, vol. 31, no. 9, pp. 2474–2485, 1988. [Online]. Available: <https://aip.scitation.org/doi/abs/10.1063/1.866601>
- [65] W. von Funck, T. Weinkauff, H. Theisel, and H.-P. Seidel, "Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 6, pp. 1396–1403, November - December 2008. [Online]. Available: <http://tinoweinkauff.net/>
- [66] F. Hussain and K. Duraisamy, "Mechanics of viscous vortex reconnection," *Physics of Fluids*, vol. 23, no. 2, p. 021701, 2011.

- [67] P. Berg, C. Roloff, and O. B. et al., "The computational fluid dynamics rupture challenge 2013phase ii: Variability of hemodynamic simulations in two intracranial aneurysms," *ASME J Biomech Eng*, vol. 137, no. 12, pp. 121 008–121 021, Dec 2015.
- [68] G. Janiga, P. Berg, S. Sugiyama, K. Kono, and D. Steinman, "The computational fluid dynamics rupture challenge 2013—phase i: Prediction of rupture status in intracranial aneurysms," *American Journal of Neuroradiology*, vol. 36, no. 3, pp. 530–536, 2015. [Online]. Available: <http://www.ajnr.org/content/36/3/530>
- [69] H. ster and R. Love, "The convergence of the weiszfeld algorithm," *Computers & Mathematics with Applications*, vol. 40, no. 4, pp. 443 – 451, 2000. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0898122100001723>
- [70] B. Walter, K. Bala, M. Kulkarni, and K. Pingali, "Fast agglomerative clustering for rendering," in *2008 IEEE Symposium on Interactive Ray Tracing*. IEEE, aug 2008. [Online]. Available: <https://doi.org/10.1109/2Frt.2008.4634626>
- [71] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional spaces," in *Proceedings of the 8th International Conference on Database Theory*, ser. ICDDT '01. London, UK, UK: Springer-Verlag, 2001, pp. 420–434. [Online]. Available: <http://dl.acm.org/citation.cfm?id=645504.656414>
- [72] L. Shi and G. Chen, "Metric-based curve clustering and feature extraction in flow visualization," in *2017 IEEE Conference on Computer-Aided Design and Computer Graphics short paper*, 2017.
- [73] L. Zhang, D. Nguyen, D. Thompson, R. Laramee, and G. Chen, "Enhanced vector field visualization via lagrangian accumulation," *Computers & Graphics*, vol. 70, pp. 224 – 234, 2018, cAD/Graphics 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0097849317301073>
- [74] D. G. Kendall, "A survey of the statistical theory of shape," *Statistical Science*, vol. 4, no. 2, pp. 87–99, 1989. [Online]. Available: <http://www.jstor.org/stable/2245331>
- [75] S. Salvador and P. Chan, "Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms," in *16th IEEE International Conference on Tools with Artificial Intelligence*, Nov 2004, pp. 576–584.
- [76] L. J. O'Donnell, C.-F. Westin, and A. J. Golby, "Tract-based morphometry for white matter group analysis," *Neuroimage*, vol. 45, p. 832–844, 2009.
- [77] S. Born, M. Markl, M. Gutberlet, and G. Scheuermann, "Illustrative visualization of cardiac and aortic blood flow from 4d mri data," in *2013 IEEE Pacific Visualization Symposium (PacificVis)*, Feb 2013, pp. 129–136.
- [78] C. Hurter, S. Puechmorel, F. Nicol, and A. Telea, "Functional decomposition for bundled simplification of trail sets," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 500–510, Jan 2018.
- [79] N. A. Yousri, M. S. Kamel, and M. A. Ismail, "A novel validity measure for clusters of arbitrary shapes and densities," in *2008 19th International Conference on Pattern Recognition*, Dec 2008, pp. 1–4.
- [80] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2-3, pp. 107–145, Dec. 2001. [Online]. Available: <https://doi.org/10.1023/A:1012801612483>
- [81] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Dec. 2010. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1756006.1953024>
- [82] D. Marghescu, "Evaluation of projection techniques using hubert's gamma statistics evaluation of projection techniques using hubert's gamma statistics," in *Proceedings of IADIS European Conference on Data Mining, 5-7 July, 2007, Lisbon, Portugal*. IADIS Press, 2007, pp. 192–197.
- [83] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, April 1979.
- [84] K. Xu and G. Chen, "Hexahedral mesh structure visualization and evaluation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 25, no. 1, pp. 1173–1182, Jan 2019.
- [85] J. Tao and C. Wang, "Semi-automatic generation of stream surfaces via sketching," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 9, pp. 2622–2635, Sept 2018.
- [86] D. House, V. Interrante, D. Laidlaw, R. Taylor, and C. Ware, "Design and evaluation in visualization research," in *VIS 05. IEEE Visualization, 2005.*, Oct 2005, pp. 705–708.
- [87] D. H. Laidlaw, R. M. Kirby, C. D. Jackson, J. S. Davidson, T. S. Miller, M. da Silva, W. H. Warren, and M. J. Tarr, "Comparing 2d vector field visualization methods: a user study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 11, no. 1, pp. 59–70, Jan 2005.
- [88] A. Forsberg, J. Chen, and D. Laidlaw, "Comparing 3d vector field visualization methods: A user study," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1219–1226, Nov. 2009. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2009.126>
- [89] J. Wei, H. Yu, R. W. Grout, J. H. Chen, and K. L. Ma, "Dual space analysis of turbulent combustion particle data," in *2011 IEEE Pacific Visualization Symposium*, March 2011, pp. 91–98.



Liyu Shi is a Ph.D. student at the Department of Computer Science, University of Houston. He received a BS in computational mathematics in Xi'an Jiaotong University in 2013. His research interest is unsupervised learning in scientific visualization and flow visualization.



Robert S. Laramee Robert received a bachelors degree in physics, cum laude, from the University of Massachusetts, Amherst (Zoo Mass) and a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics and Algorithms in 2005 (Gruess Gott TUWien). From 2001 to 2006 he was a researcher at the VRVis Research Center (www.vrvis.at) and a software engineer at AVL (www.avl.com) in the department of Advanced Simulation Technologies. Currently he is an Associate Professor at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science. His research interests are in the areas of scientific visualization, information visualization, and visual analytics.



Guoning Chen is an Associate Professor at the Department of Computer Science at the University of Houston. He earned a PhD degree in Computer Science from Oregon State University in 2009. His research interests include visualization, data analytics, computational topology, geometric modeling, geometry processing and physically-based simulation. He is a member of ACM and IEEE.