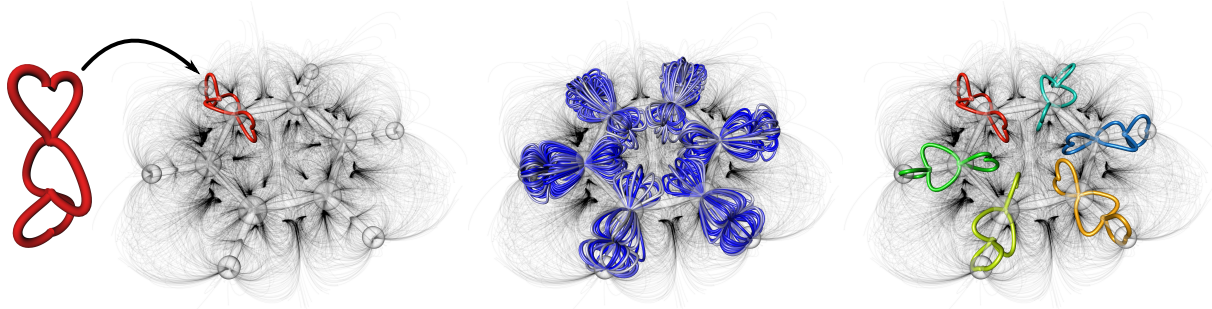# Pattern Search in Flows based on Similarity of Stream Line Segments

Z. Wang, J. Martinez Esturo, H.-P. Seidel, and T. Weinkauf

Max Planck Institute for Informatics, Saarbrücken, Germany

Figure 1: *The input to our method is a vector field and a user-defined set of stream line segments as a query pattern (● left). We find all "similar" occurrences in a location, translation, and scale-invariant way (● middle). A representative of each cluster is shown on the right.*

Abstract

We propose a method that allows users to define flow features in form of patterns *represented as sparse sets of stream line segments. Our approach finds "similar" occurrences in the same or other time steps. Related approaches define patterns using dense, local stencils or support only single segments. Our patterns are defined sparsely and can have a significant extent, i.e., they are integration-based and not local. This allows for a greater flexibility in defining features of interest. Similarity is measured using intrinsic curve properties only, which enables invariance to location, orientation, and scale. Our method starts with splitting stream lines using globally-consistent segmentation criteria. It strives to maintain the visually apparent features of the flow as a collection of stream line segments. Most importantly, it provides similar segmentations for similar flow structures. For user-defined patterns of curve segments, our algorithm finds similar ones that are invariant to similarity transformations. We showcase the utility of our method using different 2D and 3D flow fields.*

Keywords: Scientific Visualization, Flow Visualization

1 Introduction

The visualization and analysis of vector fields is of major importance for various scientific disciplines. Among the different classes of vector field visualization techniques, geometry-based techniques are well-established [MLP*10]. They rely on integral curves such as stream lines representing integrated flow behavior.

However, line-based flow visualizations face certain challenges: for instance, if applied to very complex data sets, they can quickly lead to cluttered visualizations. In particular, this is a problem for 3D flows. Figure 1 shows such an example, where the gray stream lines of the 3D vector field occlude each other to an extent that renders the entire visualization

almost illegible. The possibly existing structures within this field are lost due to visual clutter.

Furthermore, a stream line is a domain-wide integrated entity, but very often not all of its parts are equally important: for some applications, the part of a stream line in the vicinity of a vortex or critical point is more important than the part running through a region of laminar flow. However, ultimately the definition of what constitutes a "flow feature" depends on the specific application and the visualization target of the domain expert.

The method presented in this paper empowers the user to define complex flow features. We propose an example-based pattern retrieval approach: users are able to specify interesting flow features as *patterns* that are constructed of stream line *segments*, i.e., parts of stream lines. In contrast

to previous work, our method supports patterns represented by *multiple* line segments, which increases the flexibility and expressiveness of the specified patterns. This way, it is possible to specify even complex flow patterns such as the one shown in Figure 1. Patterns are matched with the vector field and successful matches of the example pattern are presented to the user. We formulate the matching to be invariant to similarity transformations, such that matched patterns are found independently of their location, orientation, or scale. In addition, pattern occurrences can be found in the same data set, in a different time step of the same data set, or in a different data set.

At its core, our formulation of flow pattern retrieval requires suitable steam line segmentations as well as measures for segment similarity, for which we propose possible solutions. Suitable stream line segmentations should allow the convenient selection of flow feature regions. We relate flow features to flow curvature, such that segment boundaries are naturally found at minima of curvatures. In addition, segmentations need to be consistent along multiple scales to allow scale-invariant pattern retrieval. We propose a new stream line segmentation scheme that is based only on intrinsic curve properties and fulfills all of these requirements. In addition, our stream line segmentation scheme and similarity estimates are valuable on their own rights and can also be use, e.g., for flow clustering applications.

Section 2 reviews approaches that are most related to our work. In the following two sections, we present details of our two main contributions: the *consistent intrinsic stream line segmentation* and *segment similarity estimation* (Section 3) and the *segmentation-based flow feature pattern retrieval* (Section 4). Section 5 validates our algorithm through several examples. Section 6 examines the efficiency of the proposed method by conducting several experiments and we discuss properties of our approach in Section 7.

2 Related Work

Stream line-based flow visualization is a well-research concept in flow visualization. A thorough review is beyond the scope of this paper and we refer to the recent survey by McLoughlin et al. [MLP*10] for an overview. In our work, we propose a new segmentation-based approach for intrinsic stream line similarity estimation that we apply to the problem of flow pattern search.

Stream Line Similarity and Clustering. A variety of approaches for similarity estimation of stream lines, in particular, and curve sets, in general, is known in the literature. Estimations generally differ in the consideration of extrinsic or intrinsic curve properties and in the discretization of these estimates. Similarities are most often required for clustering of, e.g., DTI fiber bundles [CZCE08, JDL09]. In the flow visualization context, Rössl and Theisel [RT12] perform spectral stream line clustering based on lower-dimensional embeddings guided by extrinsic stream line similarity estimates,

e.g., Hausdorff distances. Similarly, Brun et al. [BKP*04] also measure similarity in a transformed feature space. Li et al. [LWS13] propose a bags-of-feature method encoding distributions of intrinsic stream line properties and spatial relationships for similarity computations. McLoughlin et al. [MJL*13] use binned intrinsic curve properties and propose stream line similarities estimations at different resolutions. None of these methods considers shape dependent stream line *segmentations* for increased accuracy of estimated similarities. Distribution-based stream line segmentations and dynamic time warping-based stream line similarity estimations are proposed by Lu et al. [LCL*13]. Whereas their approach compares whole segmented stream lines in a scale-dependent way, we propose to measure scale-invariant similarity of better distinguishable stream line segments instead. Given a query curve, all stream line similarity estimates [RT12, LWS13, MJL*13, LCL*13] enable users to retrieve single similar stream lines in a straightforward way. Compared to similarity estimations of single curves, the problem of finding more complex flow patterns turns out to be more complex.

Flow Pattern Search. Flow patterns are generally considered as more complex distinguishable flow structures that are usually defined as query patterns by the user. Flow structures are either defined directly in terms of stencils of vector field quantities or by geometric structures like integral curves, partial integral curves, or user-sketched curves. Ebling and Scheuermann [ES03, ES06] and Heiberg et al. [HEWK03] apply convolution with dense idealized filter masks to detect predefined flow patterns. Schlemmer et al. [SHM*07] define moment-invariant pattern descriptors by dense circular vector field stencils for retrieval of flow patterns. Recently, this approach is extended by Bujack et al. [BHSE14] to include moment normalization to obtain descriptors for pattern retrieval of different scales and orientations. Both methods are restricted to 2D flows only. Single user sketched 2D curves are used as query patterns by Wei et al. [WWYM10], for which similar occurrences are retrieved from 3D line fields. As the query sketch represents a projection of the query curve, retrieval might be ambiguous for 3D stream lines. They perform no segmentation and similarity is measured by string edit distances. More recently, Tao et al. [TWS14] also propose a retrieval method based on string edit distances. Stream lines are segmented into distinctive regions to which characters of a data-dependent alphabet are assigned to enable partial stream line matching of consecutive segments.

None of these methods allows the definition of flow feature patterns in form of *multiple* stream lines or stream line segments. We now continue to present our intrinsic stream line segmentations to approach this problem.

3 Intrinsic Stream Line Segmentation

Our approach is motivated by the observation that long stream lines often pass through several mutually distinct flow features of the underlying vector field, e.g., through

multiple vortical regions. An independent analysis and retrieval of these distinct features is not possible by considering whole stream lines. Instead, our method considers spatially restricted parts of stream lines in form of shorter *stream line segments*, which can be combined to form sets of segments that represent *flow feature patterns*.

Notation. We make use of the following formal concepts: let $\mathbf{v}(\mathbf{x})$ denote a steady two or three dimensional vector field. We consider its stream lines as parametric curves $\mathbf{c}(t) = \mathbf{x}_0 + \int_0^t \mathbf{v}(\mathbf{c}(u)) \, du$, which are sampled by a curvature-based importance sampling process (see [WT02]), and later sparsified by discarding up to 90% redundant curves by checking the Hausdorff distance of any two curves (see [RT12]). We partition stream lines $\mathbf{c}$ into disjoint *stream line segments* $\mathbf{s}_i(t)$, and denote the length of $\mathbf{s}_i$ by l_i.

3.1 Globally Consistent Segmentation of Stream Lines

We identify three requirements a globally consistent segmentation has to satisfy for our application: first, a segmentation should be *feature preserving* in that all segments shall preserve the important features of the given set of curves. In general, long and sharp arcs of stream lines are considered to be important and significant curve features, while short and straight curves are less important. Second, a segment should be *distinct* enough to describe a complete feature. For instance, a circle should not be separated into two semi-circles. The third property requires a segmentation to be *consistent* in that segments, which describe similar flow features, should have similar shapes. This implies that the segmentation should be invariant to translation, rotation, scaling, and reflection, i.e., invariant to similarity transformations. These three properties are illustrated in Figure 2. Our segmentation only relies on stream line curvatures $\kappa_{2/3}(t)$. Although 2D and 3D curve curvatures are defined differently, i.e., κ_2 are signed, while κ_3 are always positive, our segmentation scheme supports 2D and 3D curves in a unified way. Stream line segmentation proceeds in two phases, curvature-based splitting and subsequent segment merging, and we continue to describe both in more detail.

Segment Splitting. Both curvature estimations κ_2 and κ_3 differ in their signedness. Hence, we consider absolute local curvatures $\hat{\kappa}(t) = |\kappa_{2/3}|$ for a unified stream line segmentation scheme that is applicable for both two and three dimensional stream lines. Vector field features are usually coupled to high absolute stream line curvatures (see, e.g., [MJL*13]). Therefore, to obtain feature-preserving and distinct segmentations, points of absolute local curvature minima that bound these high curvature regions are candidates for possible segment boundaries. We call these segments bounded by consecutive absolute local curvature minima *minimal segments*, which are the initial building blocks of the final segmentation and will not be split further.
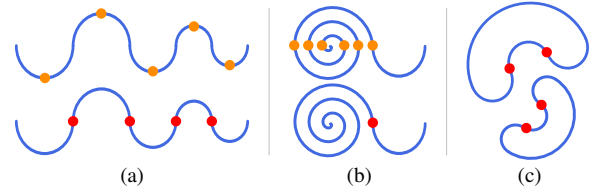
Figure 2: *Curve Segmentation. Our segmentation scheme splits stream lines in a globally consistent way at (●) (alternative, less suited split locations are colored as (●)). Shown examples illustrate different properties of our segmentation, i.e.,* feature preservation *(a),* feature distinction *(b), and segmentation* consistency *w.r.t. location, orientation, and scale (c).*


Figure 3: *Segment Merge Criteria. Pre-merge segment boundaries are colored (●), and two different average segment orientations are colored (●) and (●). (a) A pair of segments is mergeable if they both have similar average orientations. (b) A triplet of segments is mergeable if the center segment (●) has a low average total curvature compared to its neighboring segments, which have similar average orientations.*

Segment Merging. We merge neighboring segments based on two segment properties: total segment curvature and average segment orientation. Both properties are scale-invariant. The *total segment curvature* $\hat{\kappa}_i$ is given by

$$\hat{\kappa}_i = \int_{t_i}^{t_{i+1}} \hat{\kappa} \, ||\dot{\mathbf{c}}|| \, dt \ . \tag{1}$$

Along each segment, the orthonormal Frenet-Serret frames $(\mathbf{t}(t), \mathbf{n}(t), \mathbf{b}(t))$ are given by the tangent, normal, and binormal directions, respectively. We observe that along a minimal segment the variation of bi-normal directions is usually small. Therefore, we assign each segment an *average orientation* $\bar{\mathbf{b}}_i$ based on its average bi-normal direction.

Our algorithm for merging of segments consists of growing segments of low total curvature with neighboring segments, if they are merge-compatible. Compatibility is tested in two phases based on two criteria: first, two neighboring segments are mergeable if they have similar average orientations, i.e., if the angle $\alpha_i = \angle(\bar{\mathbf{b}}_i, \bar{\mathbf{b}}_{i+1})$ is smaller than a user-specified upper bound α. Second, if one segment has a low total curvature, i.e., $\hat{\kappa}_i < \beta$ for a user-specified upper bound β, it is mergeable with both of its neighboring segments if these two segments have similar average orientations w.r.t. α. Figure 3 illustrates two examples of the criteria. The merging algorithm iteratively processes segments based on a priority queue that is ordered by the total segment curvature such that segments of lowest curvature are processed first.
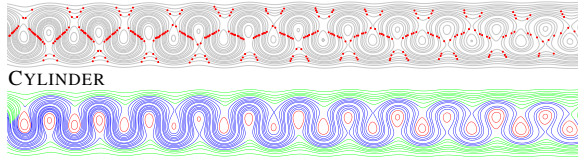
Figure 4: *Similarity-based Clustering. Using the consistent segmentation of the* CYLINDER *flow (top), a clustering of segments based on pairwise intrinsic segment similarities is computed (bottom). The shown three clusters consist of approximately laminar flow segments (●), highly curved segments (●), and circular flow segments (●).*

Figure 6: *Pattern Retrieval Overview. (a) A flow feature pattern is a set of user-selected segments (●) with one distinguished root segment (●). (b) For* global *alignment, segments similar to the root segment are found using scale-invariant intrinsic similarity (● left), then all pattern segments are transformed to their vicinity by a fitted similarity transformation (right). (c) For* local *alignment, all transformed pattern segments are matched with the local data segments to detect matched patterns (●).*

3.2 Intrinsic Similarity of Stream Line Segments

Based on our consistent curve segmentation scheme, we propose a general scale-invariant method for intrinsic curve segment comparison.

First, we discretize the continuous intrinsic curve properties like curvatures along each segment into $n > 0$ uniformly sized bins. The parameter n steers the profile resolution and accuracy. In all our experiments, we observe that a value of $n = 40$ is usually sufficient to enable accurate segment comparison, e.g., for pattern retrieval. For comparability, we scale-normalize each profile by the curve lengths l_i.

To measure the intrinsic similarity of a pair of curve segments, we employ hEMD [PW09], which is a generalization of the *Earth Movers Distance* (EMD), for the comparison two scale-normalized profiles. hEMD is a cross-bin measure that is more robust w.r.t. local deformations and also a well-defined metric for our setting of unequal total profile sums.
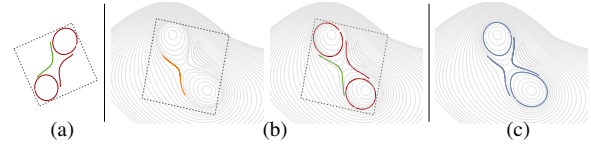
In order to obtain similarity transformation invariance for 2D curves, we need to compute the minimum of four distance measures, i.e., two for inverting the curve traversal order, and two for flipping the sign of the curvature. For 3D segment similarity estimation, we combine differences in unsigned curvature κ_3 and torsion τ as $d_{\kappa_3} + w_\tau d_\tau$ of individual hEMD profile distances in curvature d_{κ_3} and torsion d_τ. A weight parameter $w_\tau < 1$ is chosen to reduce the influence of torsion to the final similarity estimation. Similar to the 2D case, to evaluate the similarity of two 3D segments, four hEMD evaluations are required, i.e., two for inverting the traversal order, and two for flipping the sign of the torsion.

4 Pattern Search

In this section, we propose an example-based flow pattern search approach for the detection of similar flow feature patterns given a query pattern. We formulate it independently of the curve dimension and the algorithm is applicable for both 2D and 3D curves.

4.1 Pattern Definition

Patterns $\mathcal{P} \subset \mathcal{S}$ are given by a subset of segments from the set of all stream line segments $\mathcal{S}$. Example query patterns are selected by the user to define flow features for pattern retrieval. Usually, a flow feature pattern consists of distinctive elements of different types of flow features. For instance, a saddle point surrounded by two vortices can be described using the following stream line segments: two arcs coupled with two circular stream lines at either side, see Figure 6. Intrinsically similar occurrences of the pattern can be retrieved from other parts of the domain, from other points in time, or even from other data sets.

4.2 Pattern Retrieval

Given a query pattern, we search for differently scaled and geometrically compatible pattern locations in the search space of all segments. Pattern retrieval consists of two consecutive *global* and *local* phases. The procedure is illustrated in Figure 6 and we continue to present its details.

Global Pattern Alignment. Global pattern alignment requires the localization of candidate patterns and the computation of the similarity transformations that align the query pattern with candidate matches.

We define the segment of highest end point distance as the *root segment* $\mathbf{r}(s) \in \mathcal{P}$ to perform similarity computations. For pattern search in 3D, we additionally require the root segment to have non-vanishing curvature. Using the root segment, we find a set $\mathcal{Q}$ of similar segments $\mathbf{q}(s) \in \mathcal{Q} \subset \mathcal{S}$ using the scale and orientation-invariant segment similarity of Section 3.2. To increase the matching performance, for a given root segment, users can reduce the size of $\mathcal{Q}$ by prescribing a percentage p of the considered range of similarity values. In our experiments, $p = 10\%$ turns out to be sufficient. Given two similar segments $\mathbf{r}$ and $\mathbf{q}$, we resample the shorter curve to have the same number $|\mathbf{r}| = |\mathbf{q}| = m$ of vertices as the longer one. The optimal alignment $\mathbf{q}_k = c\,\mathbf{R}\,\mathbf{r}_k + \mathbf{t}$ consisting of three fitted similarity transformation components, i.e., a translational part $\mathbf{t}$, a rotational part $\mathbf{R}$, and a scaling factor c, is then computed by the technique proposed by Arun et al. [AHB87].
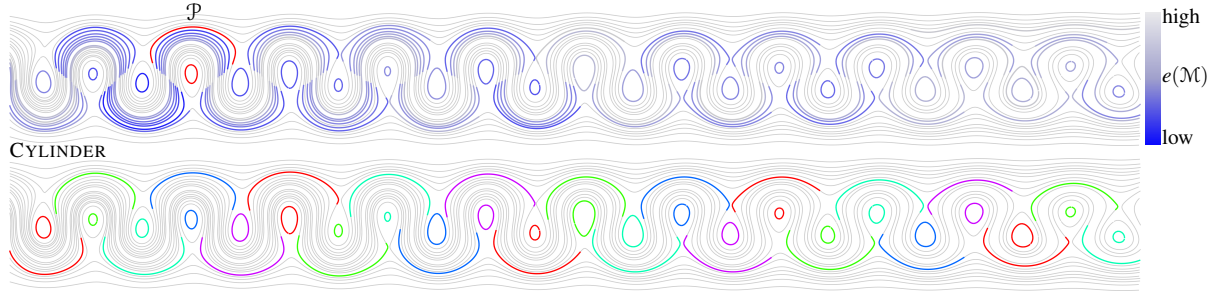
Figure 5: *Pattern Retrieval in the* CYLINDER *Flow. In a single time step of the flow behind a circular* CYLINDER *obstacle (not shown) a flow pattern* $\mathcal{P}$ (●) *is selected. Our pattern retrieval evaluates the geometric matching costs* $e(\mathcal{M})$ *of local candidate occurrences of the pattern at different locations, orientations, and scales (top). Clustering of locally similar candidate occurrences yields representative and distinctive pattern matches (bottom, differently colored). The consistent segmentation of this flow is shown in Figure 4.*

Local Pattern Matching. Consecutively, in the local pattern alignment step, each geometrically similar segments are considered and a pattern match is found if every segment can be matched to a compatible segment.

Note that, for local pattern matching, we perform an extrinsic similarity estimation, as for pattern search extrinsically matching segment configurations are required: for two segments **p** and **q** with vertices $\mathbf{p}_i$ and $\mathbf{q}_j$ (not necessarily of equal number), we measure *extrinsic* shape similarity using the symmetric *Chamfer distance* $e_c(\mathbf{p}, \mathbf{q})$ given by

$$e_c(\mathbf{p} \to \mathbf{q}) = \frac{1}{|\mathbf{p}|} \sum_i \min_j ||\mathbf{p}_i - \mathbf{q}_j|| \qquad (2)$$

$$e_c(\mathbf{p}, \mathbf{q}) = \max(e_c(\mathbf{p} \to \mathbf{q}), e_c(\mathbf{q} \to \mathbf{p})) . \qquad (3)$$

The Chamfer distance is successfully applied for other shape matching problems [Gav07] and measures the average sum of closest point distances. To allow slight variations in segment shape and position, for pattern matching we use the *segment matching cost* function

$$e(\mathbf{p}, \mathbf{q}) = e_c(\mathbf{p} - \bar{\mathbf{p}}, \mathbf{q} - \bar{\mathbf{q}}) + w_e ||\bar{\mathbf{p}} - \bar{\mathbf{q}}|| \qquad (4)$$

given as a combination of mass centered extrinsic shape similarity and segment distance expressed by the distance of the respective centers of mass $\bar{\mathbf{p}}$ and $\bar{\mathbf{q}}$. The weight w_e allows to balance between the required shape similarity and the allowed segment distance and it can be choose to be $w_e = 1$ for equal weighting.

To locally match a pattern to a candidate match position, let $\mathcal{P}'$ denote the set of segments transformed by the fitted similarity transformation. The set of candidate match segments $\mathcal{C} \subset \mathcal{S}$ of the underlying flow are all the segments for which at least a single vertex is inside the transformed bounding box of the pattern. The set of matched segments is computed as the set

$$\mathcal{M} = \left\{ \hat{\mathbf{q}}_i \mid \forall \mathbf{p}'_j \in \mathcal{P}' : \hat{\mathbf{q}}_i = \underset{\mathbf{q}_i \in \mathcal{C}}{\arg\min} \; e(\mathbf{p}'_j, \mathbf{q}_i) \right\} \qquad (5)$$

of most similar candidate segments to the transformed pattern segments. The total *pattern matching cost* $e(\mathcal{M}) = \sum_i e(\mathbf{p}'_i, \hat{\mathbf{q}}_i)$ is then given by the sum of individual minimal segment matching costs.

As a single flow feature should only be represented by a single matching pattern, we cluster multiple close matchings in a straightforward way by clustering the centers of mass of their bounding boxes.

5 Validation

In this section, we use the well-known 2D flow behind a CYLINDER that develops a von Kármán vortex street behind an obstacle to validate our algorithm.

In Figure 4 (bottom), we illustrates a hierarchical clustering of all the stream line segments of the CYLINDER Flow based on pairwise intrinsic segment similarities. The consistent segmentation of this flow is shown in Figure 4 (top). For clustering, we uses Ward's minimum variance algorithm [War63]. All the segments are grouped into three clusters: laminar flow segments, highly curved segments, and circular flow segments. Note that intrinsically similar segments are grouped to clusters at different scales, which shows the scale-invariance of our similarity estimation.

In Figure 5, we apply our pattern retrieval approach to a single time step of the CYLINDER flow. For a given user-selected flow feature pattern $\mathcal{P}$, we show the resulting pattern matching costs $e(\mathcal{M})$ in the upper part. Pattern matching costs in the vicinity of the original pattern are high due to the similarity of the surrounding repetitive flow features at similar scale. Matching costs slightly increase away from the obstacle due to decreased extrinsic similarity to $\mathcal{P}$. The bottom figure shows the distinctively clustered pattern representatives after aggregating close matchings. The repetitive flow feature is well represented in the matching results.

In another validation example, we apply a pattern from one time step to all other time steps. We demonstrate this in Figure 7: given the pattern $\mathcal{P}$ selected in one time step of the previous CYLINDER example (Figure 5), we match it to all occurrences at *every* time step of the time-dependent CYLINDER flow. Shown are the matched patterns in all following time steps, which we visualize in space-time domain. The result is consistent with the single time step result and the individual pattern evolution is well-represented. Note that
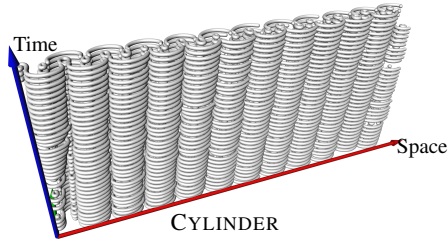
Figure 7: *Time-dependent Pattern Search. For the time-dependent 2D* CYLINDER *flow, we perform pattern retrieval for the pattern* $\mathcal{P}$ *selected in the time-step shown in Figure 5. The consistent matching results are visualized in space-time domain.*
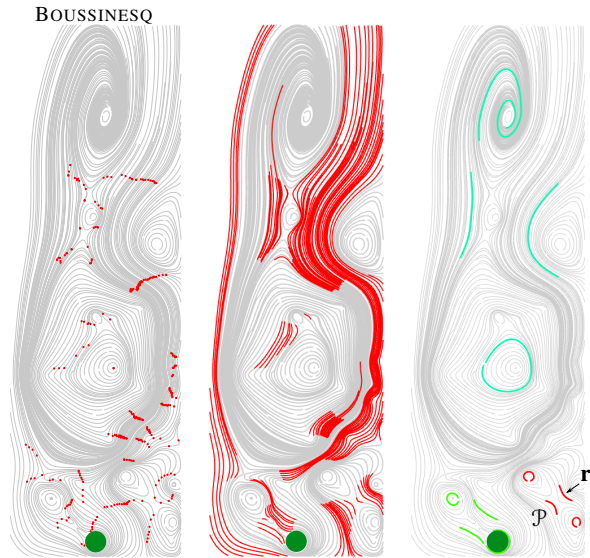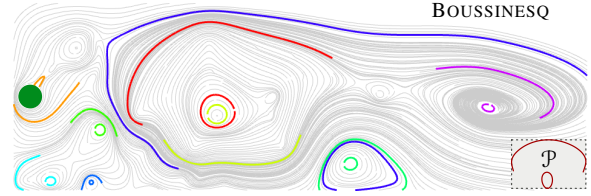

Figure 8: BOUSSINESQ *Pattern Search. The* BOUSSINESQ *flow represents the advective mass transport induced by a circular heat source (●). In the segmented flow (left), we search for occurrences of the selected pattern* $\mathcal{P}$ *(●, bottom right). For the root segment* **r** *(bottom right), the middle image shows the most similar segments of the data set that are used for matching. The pattern is matched to two different occurrences (right, differently colored) at different locations, orientations, and scales.*

this example does not focus on *tracking* of individual segments over time. Rather, it demonstrates the ability of our approach to match the extrinsic configuration of a consistently segmented pattern from one time step to any other time step.

6 Results

We continue to present results of our pattern retrieval approach in this section. In Figure 8, we show pattern search results for the more complex simulated 2D BOUSSINESQ flow representing the Boussinesq approximation applied to


Figure 9: *Pattern Search with External Pattern. In the* BOUSSINESQ *flow, we search for the pattern* $\mathcal{P}$ *(bottom right) that is given by two external segments from the* CYLINDER *flow shown in Figure 5. The pattern is matched to nine occurrences in the* BOUSSINESQ *flow (differently colored) at various different* nested *locations, orientations, and scales.*

solve for the flow generated by a heated cylinder. We show both the segmentation result and the most similar segments to a given root curve of the user selected pattern in the same data set. The flow pattern consists of two vortical regions that are separated by a saddle-like structure. In the same data set, two occurrences are detected. They are matched at very different scales, illustrating the scale-invariance of our approach.

In the same date set illustrated in Figure 9, we exemplify the use case of matching with *externally* defined patterns: using the 2D flow pattern defined by consistently segmented segments in the CYLINDER flow of Figure 5, we detect nine matching occurrences in the BOUSSINESQ flow. Note that matchings are found at various different locations, orientation, and scales. As long as a query pattern consists of consistently segmented segments it can be used as an *external* pattern in our method. In particular, external patterns need not conform to the scale of the data set due to the invariance to similarity transformations of our method. In addition, this result demonstrates that our approach also retrieves *nested* matches, i.e., pattern occurrences at different scales but at same locations. Nested patterns can be interpreted to give a multi-scale representation of a particular flow pattern. They are supported by our method due to the sparseness of our pattern definition. Note that matching of nested patterns is usually not supported by pattern matching approaches that are defined by dense stencils [ES06, HEWK03, SHM*07, BHSE14], as this would require self-similar stencils.

In Figure 1, we illustrate the application of our approach to 3D pattern retrieval in the electrostatic field around a BENZENE molecule [SS96]. The specified complex pattern is matched at six accumulation points corresponding to the six-fold molecule symmetry. This example demonstrates that for 3D flows matched patterns are also detected in a rotational-invariant way by our method.

The DELTAWING flow in Figure 10 is a simulated field around a triangle-shaped airplane, courtesy of Markus Rütten (DLR). By selecting a straight stream line segment that has a spiraling segment at the tip of one of the two vortices as the flow feature pattern, our method detects similar stream lines that enter one of both vortices. We only show segments
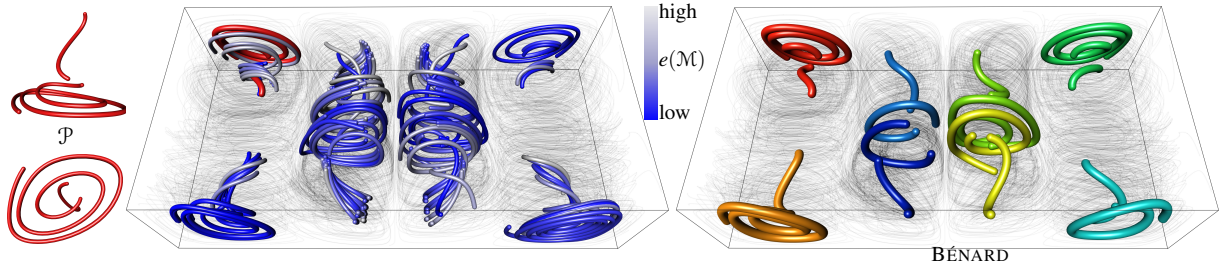
Figure 11: *3D Pattern Search in the* BÉNARD *Flow. For the user-selected pattern* $\mathcal{P}$ *(● left) in the Rayleigh-*BÉNARD *convection flow, the pattern matching costs (middle, low costs matches are rendered with thicker lines) indicate eight locations of increased pattern occurrences. All eight distinctive pattern matches are found by clustering these matches for the retrieval result (right).*
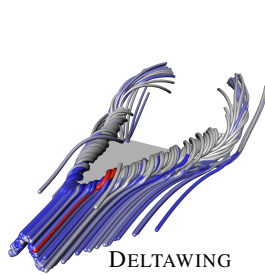
Figure 10: *3D Pattern Search in the* DELTAWING *Flow. The selected pattern* $\mathcal{P}$ *(●) consists of a straight segment combined with spiraling segment at the tip of one vortex. Segments with small matching costs (●) are detected close to similar regions entering both vortices.*

Data set	$|\mathcal{L}|$	$|\mathcal{S}|$	$|\mathcal{Q}|$	$|\mathcal{P}|$	SIMI (s)	PSEARCH (s)
CYLINDER	55	400	400	2	0.6	0.3
BOUSSINESQ	172	473	154	4	0.9	1.7
BENZENE	7832	9116	4019	6	162	137
BÉNARD	266	8747	547	2	57	119
DELTAWING	302	1700	1646	2	12	1.6

Table 1: *Timings. For each data set, we list the number integrated stream lines $|\mathcal{L}|$, number of total segments $|\mathcal{S}|$, the number of considered match candidates $|\mathcal{Q}|$, the number of pattern segments $|\mathcal{P}|$, root curve intrinsic similarity computation (*SIMI*) as well as the local and global phases of the pattern search with match clustering (*PSEARCH*).*

having this characteristic and color code their similarity to the pattern.

The Rayleigh-BÉNARD flow in Figure 11 is a simulated data set of fluid motion as the result of thermal convection of a heated and cooled boundaries, obtained using the software NaSt3DGP (University of Bonn). The selected pattern consists of a vortical part with an orthogonally aligned segment. For each segment of the search space the middle image shows the individual pattern matching costs, which results in eight locations of increased pattern occurrences. Segment clustering results in eight representative and distinctive pattern matches.

Table 1 summarizes the used number of segments and processing time of all examples of this section. Processing times were measured with a parallel implementation on an

Intel Core i7-4770K 3.5GHz quad core system. Our consistent stream line segmentation is a very efficient operation even for a high number of stream lines: for all tested data sets, segmentation time is less than 0.03 seconds. Among all operations, the similarity computation of the root curve to the search space segments is one of the most expensive steps of our method. In fact, the costs are not unexpected, as this operation effectively corresponds to a *global* and *scale-invariant* segment matching. Note that other methods evaluating distances between discretized intrinsic property profiles [MJL∗13, LCL∗13] have similar overall complexity. Slightly higher runtimes are caused by our usage of the more general and accurate, but also more expensive hEMD distance estimation, if compared to standard χ^2 or EMD distances. Similar to similarity estimations, pattern search performance depends on the number of tested segments and their vertex count, which we found to be highly data set-dependent. Still, all operations can be parallelized in a straightforward way for increased performance.

7 Discussion

In our approach, we design the retrieval of flow feature patterns based on segmentations of stream lines. Compared to existing methods, this has a number of implications w.r.t. stream line similarity estimation and flow pattern matching.

For stream line similarity estimation, it leads to ill-posed similarity estimations if curves of different segment number are compared, e.g., a single-arc curve with a multi-arc curve. But in our work we emphasize the importance of only comparing compatible curve segments that usually represent a single dominant region of maximal curvature. On one hand, this restricts the spatial extend of the comparable entities, on the other hand, similarity estimations become more reliable. In fact, our method explicitly supports the definition of sparse sets of stream line segments as query patterns. This increases the flexibility of pattern definition and enables a greater range of possible pattern retrieval applications, e.g., the transfer of patterns to external data sets or the matching of nested flow patterns.

Limitations and Outlook. Although stream line segmentation enhances the reliability of segment similarity computations, we identify a number of related drawbacks. First,

as we define flow patterns as sets of stream line segments, users must stick to these segments when defining patterns. Artificially designed patterns will often not match properly. Usually this is unproblematic, since segments correspond to intuitive and distinctive flow regions due to the consistent curvature-based segmentation. However, certain types of flow patterns are harder to describe this way: an example is the combination of a vortical region with a *straight* curve of limited extend in its vicinity, because straight curves will not be segmented into individual segments due to the absence of curvature. It is an interesting direction for further research to identify alternative curve segmentations, e.g., hierarchical multi-resolution schemes for segmentation and partial matching approaches, which could alleviate this limitation. Since our algorithm is purely geometric, another interesting future direction is to apply it to other types of curves such as streak and time lines [WHT12], or fiber tracts [CZCE08, JDL09].

8 Conclusion

In this work, we presented a novel approach to pattern retrieval in flows that is based on a consistent stream line segmentation. Flow patterns are defined by sparse sets of stream line segments. This provides flexibility for their definition. They are matched independently of position, location, and scale to the same data set, a different time step, or even a different data set. We demonstrate the pattern retrieval effectiveness on a number of 2D and 3D data sets. Efficient pattern matching is enabled by a new stream line segmentation scheme that is solely based on intrinsic curve properties and segments intrinsically similar stream lines in a scale-invariant and feature-consistent way. Based on this segmentation, intrinsic segment similarity estimates are proposed that are invariant w.r.t. rigid or similarity transformations.

Acknowledgments

References

[AHB87] ARUN K. S., HUANG T. S., BLOSTEIN S. D.: Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence 9*, 5 (1987), 698–700. (Cited on page 4)

[BHSE14] BUJACK R., HOTZ I., SCHEUERMANN G., E.HITZER: Moment invariants for 2d flow fields using normalization. In *Proc. IEEE Pacific Visualization 2014* (2014). (Cited on pages 2 and 6)

[BKP*04] BRUN A., KNUTSSON H., PARK H.-J., SHENTON M. E., WESTIN C.-F.: Clustering fiber traces using normalized cuts. In *MICCAI*. Springer, 2004, pp. 368–375. (Cited on page 2)

[CZCE08] CHEN W., ZHANG S., CORREIA S., EBERT D. S.: Abstractive representation and exploration of hierarchically clustered diffusion tensor fiber tracts. *Comput. Graph. Forum (Proc. EuroVis) 27*, 3 (2008), 1071–1078. (Cited on pages 2 and 8)

[ES03] EBLING J., SCHEUERMANN G.: Clifford convolution and pattern matching on vector fields. In *Proc. IEEE Visualization* (2003), pp. 193–200. (Cited on page 2)

[ES06] EBLING J., SCHEUERMANN G.: Segmentation of flow fields using pattern matching. In *Proc. EuroVis* (2006), pp. 147–154. (Cited on pages 2 and 6)

[Gav07] GAVRILA D. M.: A bayesian, exemplar-based approach to hierarchical shape matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 8 (2007), 1408–1421. (Cited on page 5)

[HEWK03] HEIBERG E., EBBERS T., WIGSTROM L., KARLSSON M.: Three dimensional flow characterization using vector pattern matching. *IEEE Transactions on Visualization and Computer Graphics 9*, 3 (2003), 313–319. (Cited on pages 2 and 6)

[JDL09] JIANU R., DEMIRALP C., LAIDLAW D. H.: Exploring 3d dti fiber tracts with linked 2d representations. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (2009), 1449–1456. (Cited on pages 2 and 8)

[LCL*13] LU K., CHAUDHURI A., LEE T., SHEN H., WONG P. C.: Exploring vector fields with distribution-based streamline analysis. In *IEEE Pacific Visualization 2013* (2013). (Cited on pages 2 and 7)

[LWS13] LI Y., WANG C., SHENE C.: Streamline similarity analysis using bag-of-features. In *Proc. SPIE* (2013), vol. 9017, pp. 90170N–90170N–12. (Cited on page 2)

[MJL*13] MCLOUGHLIN T., JONES M. W., LARAMEE R. S., MALKI R., MASTERS I., HANSEN C. D.: Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics 19*, 8 (2013), 1342–1353. (Cited on pages 2, 3, and 7)

[MLP*10] MCLOUGHLIN T., LARAMEE R. S., PEIKERT R., POST F. H., CHEN M.: Over two decades of integration-based, geometric flow visualization. *Computer Graphics Forum 29*, 6 (2010), 1807–1829. (Cited on pages 1 and 2)

[PW09] PELE O., WERMAN M.: Fast and robust earth mover's distances. In *Proc. ICCV* (2009), IEEE. (Cited on page 4)

[RT12] RÖSSL C., THEISEL H.: Streamline embedding for 3d vector field exploration. *IEEE Transactions on Visualization and Computer Graphics 18*, 3 (2012), 407–420. (Cited on pages 2 and 3)

[SHM*07] SCHLEMMER M., HERINGER M., MORR F., HOTZ I., HERING-BERTRAM M., GARTH C., KOLLMANN W., HAMANN B., HAGEN H.: Moment invariants for the analysis of 2d flow fields. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1743–1750. (Cited on pages 2 and 6)

[SS96] STALLING D., STEINKE T.: *Visualization of Vector Fields in Quantum Chemistry*. Tech. rep., ZIB Preprint SC-96-01, 1996. (Cited on page 6)

[TWS14] TAO J., WANG C., SHENE C.: Flowstring: Partial streamline matching using shape invariant similarity measure for exploratory flow visualization. In *Proc. IEEE Pacific Visualization* (March 2014). (Cited on page 2)

[War63] WARD J. H.: Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association 58*, 301 (1963), 236–244. (Cited on page 5)

[WHT12] WEINKAUF T., HEGE H.-C., THEISEL H.: Advected tangent curves: A general scheme for characteristic curves of flow fields. *Computer Graphics Forum (Proc. Eurographics) 31*, 2 (April 2012), 825–834. (Cited on page 8)

[WT02] WEINKAUF T., THEISEL H.: Curvature measures of 3D vector fields and their applications. *Journal of WSCG 10*, 2 (2002), 507–514. (Cited on page 3)

[WWYM10] WEI J., WANG C., YU H., MA K.: A sketch-based interface for classifying and visualizing vector fields. In *Proc. IEEE Pacific Visualization* (2010), pp. 129–136. (Cited on page 2)