

Exploratory segmentation of vector fields using multidimensional projection

Danilo Motta*, Maria Oliveira*, Paulo Pagliosa[†], Luis Gustavo Nonato* and Afonso Paiva*

*ICMC – USP, São Carlos

[†]FACOM – UFMS, Campo Grande

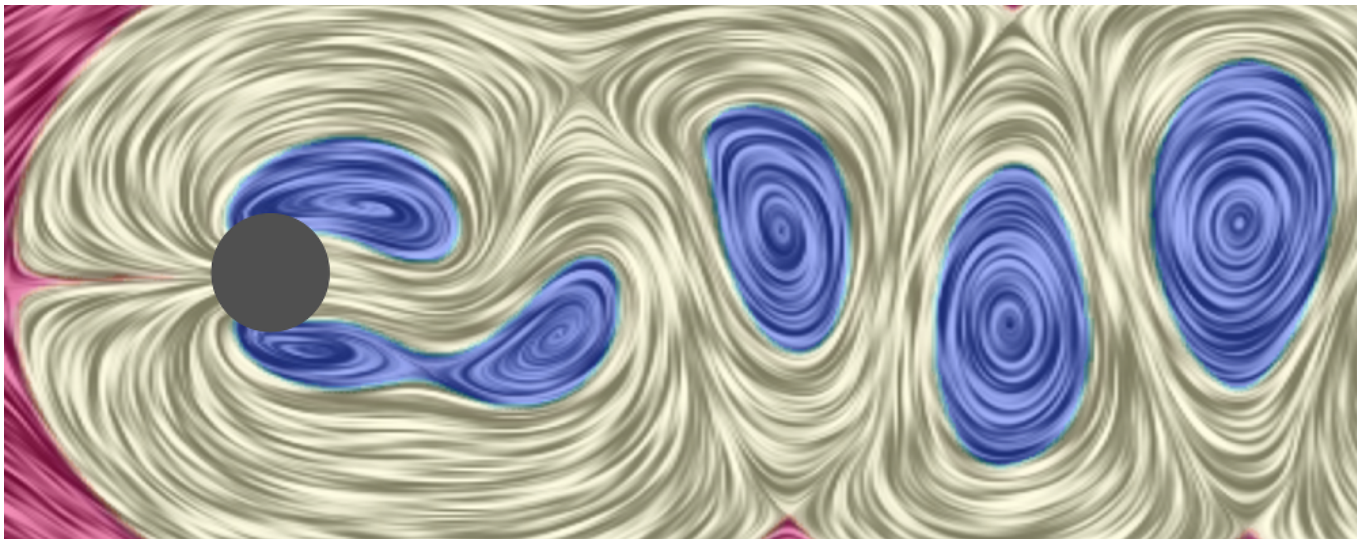


Fig. 1. Segmentation of a von Kármán vortex street generated by a cylindrical object (dark gray): our method takes as input a discrete vector field (represented by the LIC) and produces as result a classification of the cells of the underlying grid. Cells belonging to the same class define the regions of the vector field which share similar flow behavior such as vortical structures (purple) and saddle points (yellow).

Abstract—The difficulty to understand the complex behavior of vector fields makes its visual segmentation an area of constant interest in scientific visualization. In this paper, we present a novel interactive segmentation framework for discrete vector fields. In our method, the vector field domain is partitioned into multiple regions with same flow patterns. In order to accomplish this task, feature vectors are extracted from streamlines and mapped to a visual space using multidimensional projection. The interactivity with projected data in the visual space improves the results of the segmentation according to user’s knowledge. The provided results and comparisons show the flexibility and effectiveness of our framework.

Keywords—vector field; segmentation; multidimensional projection; visualization.

I. INTRODUCTION

Visualizing and analyzing the behavior of vector fields help to provide insight about the underlying physical phenomena. However, achieving a consistent interpretation of vector fields and its structures is not straightforward, leveraging the development of distinct methodologies such as flow visualization and segmentation methods [1]. In particular, vector field segmentation, which is the focus of this work, has gained great interest in the last decade, mainly motivated by the capability

of most segmentation methods in identifying regions of similar flow behavior, thus facilitating the recognition of structures such as vortices, saddle points, sources and sinks.

Vector field segmentation methods can be grouped and organized in three main categories: topology-based, geometry-based and methods based on feature spaces.

Topology-based methods rely on the detection and classification of critical points from which one can compute separatrix curves that bound the different regions of the vector field [2], [3]. Effective and robust approaches are available for two-dimensional flows [4], [5]. However, the computational implementation of these methods are quite intricate, they are computationally expensive and, with very few exceptions (e.g., [6]), they do not allow for user intervention during the segmentation process.

Geometry-based methods make use of geometric attributes of the vector field in order to identify regions with similar flow behavior. The technique proposed by Li et al. [7] is a typical representative of geometry-based vector field segmentation method, which relies on Green functions to perform a discrete Helmholtz-Hodge decomposition that drives the flow segmentation. Geometric properties of streamlines have

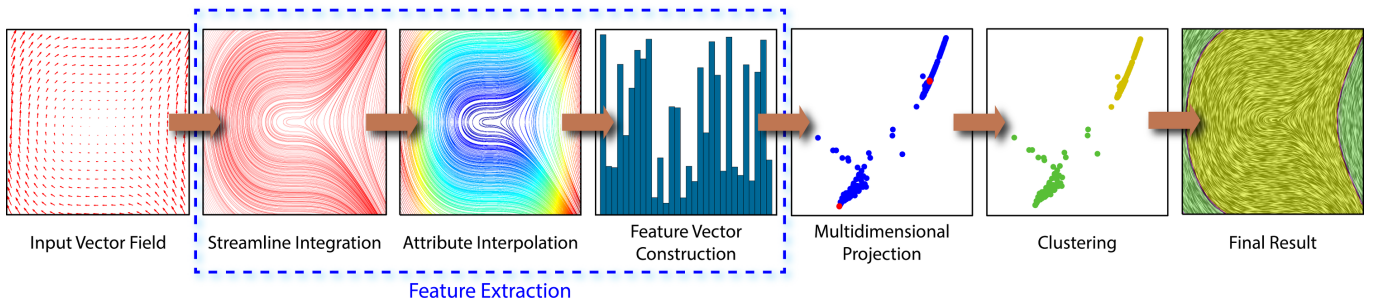


Fig. 2. Pipeline of our vector field segmentation framework.

been used by Kuhn et al. [8] to identify regions of the flow with similar streamlines. The segmentation is performed by clustering streamlines based on an energy minimization procedure. Besides being computationally intricate, Li’s and Kuhn’s methods do enable user intervention to assist the segmentation process.

Methods based on attribute space rely on features extracted from a vector field in order to identify regions of similar flow behavior. Daniels et al. [9], for instance, introduced a methodology for interactive exploration of 2D vector fields in which the whole vector field is mapped to an attribute space and then projected onto a two-dimensional visual space where the user can interact. However, this approach though, cannot accurately identify regions of similar flow. Rössl and Theisel [10] proposed a streamline segmentation method where multidimensional scaling is used to embed streamlines into a feature space. The embedding relies on the Hausdorff distances between streamlines, which demands the computation of the full distance matrix between all the streamlines sampled in the domain. User interaction is another issue in their approach.

This paper presents a novel vector field segmentation framework which comprises a set of traits difficult to be found in other methodologies. Similar to Kuhn et al. [8] and Rössl and Theisel [10] methods, we focus on streamlines to accomplish the segmentation; however, in our method the streamlines are embedded in a feature space, thus allowing the use of Euclidean distance to compute their similarity. Moreover, our method enables interactive mechanism, allowing the user to incorporate his/her knowledge about the field into the segmentation process. Therefore, our technique combines the good properties of streamline-based segmentation methods with the flexibility of interactive techniques for vector field segmentation. Figure 1 shows our framework in action, regions with same flow behavior are dyed with the same color, even if they are not geometrically close.

In summary, the main contributions of this paper are:

- This paper introduces a new mechanism to embed streamlines in attribute spaces;
- An interactive framework using multidimensional projection that allows for user intervention during the segmentation process;
- Our method can be applied to 2D and 3D vector fields.

We show the effectiveness of our method in a set of experiments and comparisons with state-of-the-art techniques.

II. THE PROPOSED APPROACH

As illustrated in Figure 2, the proposed vector field segmentation method comprises three main steps: feature extraction, multidimensional projection and clustering. In the first step, features are extracted from streamlines defined by the vector field. More precisely, each streamline is associated to a feature vector in a high-dimensional attribute space.

The high-dimensional data is then projected onto a visual space using a multidimensional projection method in the second step of the proposed pipeline. The projected data is finally clustered and the resulting clusters provide the vector field segmentation. As we clarify in the following, the advantages of clustering in visual space are twofold. The user can visualize sets of similar streamlines, which helps to identify the number of clusters as well as particular structures in the vector field. Moreover, the user can interact with the projected data, making it possible to interactively modify clusters and/or define new ones.

Next, we detail each step of our framework.

A. Feature extraction

The proposed approach relies on features computed on streamlines to segment a given vector field. The feature computation is made up of three main steps: streamline integration, attribute interpolation, and feature vector construction.

Streamline integration Let G be a grid and V be a vector field defined in each node of G ; that is, each grid node has a vector V associated to it. We sample streamlines in G so as to ensure that each cell $c_{ij} \in G$ is intersected by at least one streamline. The sampling procedure is performed as follows: starting from the top leftmost cell c_{00} we integrate a streamline starting from the center of c_{00} using Heun’s predictor-corrector method with adaptive time step. Bilinear interpolation is employed to interpolate the vector field in each integration step when the differential equation that defines the vector field is not available. After integrating the first streamline, we traverse the cells in G left-right/top-down verifying if each cell c_{ij} has been intersected by a streamline, skipping to the next cell if true. If not, that is, if c_{ij} does not intersect any streamline, we sample a new streamline from the

center of c_{ij} , performing the integration in both forward and backward directions.

At the end of the streamline sampling process, we guarantee that every cell in G is intersected by at least one streamline. Moreover, the number of streamlines integrated during the sampling process is typically much smaller than the number of cells in G , as a single streamline tends to intersect many cells and intersected cells are not sampled during the process.

Attribute interpolation Once streamlines have been sampled, differential and geometric attributes of the vector field are interpolated onto the streamlines as follows. Let S be a streamline and s_i be a point in S given by the integration process described above. We associate a set of attributes derived from the vector field to each point s_i . More precisely, curvature, magnitude, vorticity and divergence of velocity are computed in each node of the grid G and then interpolated in s_i using bicubic interpolation. Curvature and vorticity are computed as proposed by McLoughlin et al. [11] and Post et al. [1], while the divergence is given by finite differences.

Feature vector construction In order to assign a single feature vector for a streamline S , an attribute value a_i is computed at each point $s_i \in S$. These values are quantized in m bins to generate a frequency histogram. The bin intervals are determining as follows

$$[a_{min} + ih, a_{min} + (i + 1)h], \quad i = 0, \dots, m,$$

where $h = (a_{max} - a_{min})/m$ with $a_{min} = \min\{a_i\}$ and $k_{max} = \max\{a_i\}$. The resulting histogram is created concatenating the attribute histograms into a $4m$ -dimensional feature vector associated to each streamline (Figure 3). In other words, each streamline is represented by a $4m$ -dimensional point in an feature space. After applying the procedure above to all sampled streamlines, we normalize each histograms dividing it by the number of sample points in the corresponding streamline.

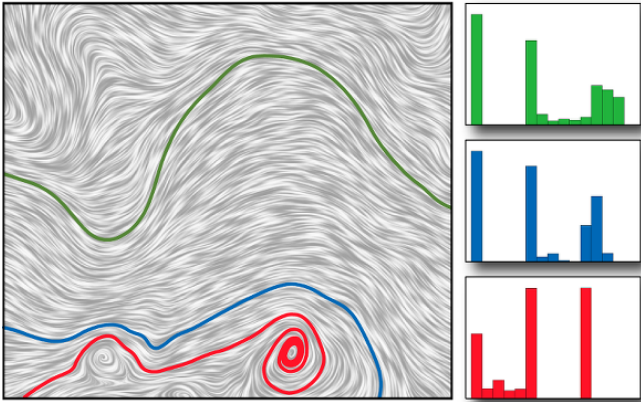


Fig. 3. Mapping each streamline to a histogram of attributes.

B. Multidimensional projection

The main advantage of converting streamlines into feature vectors embedded in a high-dimensional space is that Euclidean distance can be used to compare streamlines, thus allowing the application of well known clustering algorithms to group streamlines according to their similarity. Moreover, embedding streamlines in an attribute space also enables the use of multidimensional projection to visualize the similarity relations between the streamlines. More precisely, multidimensional projection methods such as LAMP [12] can project data from a high-dimensional attribute space to a two-dimensional visual space such that points that are close in the attribute space are projected close to each other in the visual space.

The multidimensional projection step is responsible to map high-dimensional data onto a two-dimensional visual space. In our implementation we are employing the multidimensional projection technique called LAMP (*Local Affine Multidimensional Projection*). Besides preserving neighborhood structures quite well as demonstrated in [12], the LAMP technique enables a very flexible mechanism to interactively modify the projection according to user interventions, which can be a helpful feature for manual cluster selection.

LAMP uses a set of control points to perform the mapping of a set of high-dimensional data \mathcal{X} to the visual two-dimensional space. The set of control points is typically a small subset $\mathcal{X}_S \subset \mathcal{X}$ whose counterpart \mathcal{Y}_S in the visual space is known a priori (\mathcal{X}_S can be mapped to the visual space using distance preserving optimization scheme as proposed by Tejada et al. [13]). The mapping of each instance $x \in \mathcal{X}$ to a point y in the visual space is carried out by finding the best affine transformation $y = f_x(p) = pR + t$ that minimizes

$$\sum_i \alpha_i \|f_x(x_i) - y_i\|^2 \quad \text{subject to} \quad R^\top R = I, \quad (1)$$

where the matrix R and vector t are unknowns, I is the identity matrix, $x_i \in \mathcal{X}_S$ is the i -th control point, $y_i \in \mathcal{Y}_S$ is the mapping of x_i in the visual space, and $\alpha_i = 1/\|x_i - x\|^2$ is a scalar weight. The orthogonality constraint $R^\top R = I$ enforces that the resulting affine transformation behaves like a rigid transformation, thus preserving distances as much as possible and ensuring that errors introduced during the positioning of control points are not drastically propagated during the projection step. The minimization problem (1) can be expressed in matricial form (see [12] for details):

$$\text{minimize } \|AR - B\|_F \quad \text{subject to} \quad R^\top R = I, \quad (2)$$

where $\|\cdot\|_F$ is the Frobenius norm, and A and B are matrices whose rows correspond to the coordinates of each control point in the high-dimensional and visual spaces, respectively. When a user manipulates control points in the visual space he/she also changes entries in the rows of matrix B , thus tuning the mapping R to cope with the user intervention. More precisely, when a control point is moved in the visual space the data in its neighborhood is moved together, what allows for controlling neighborhood structures in the visual space. This flexibility will be exploited in our context to enable

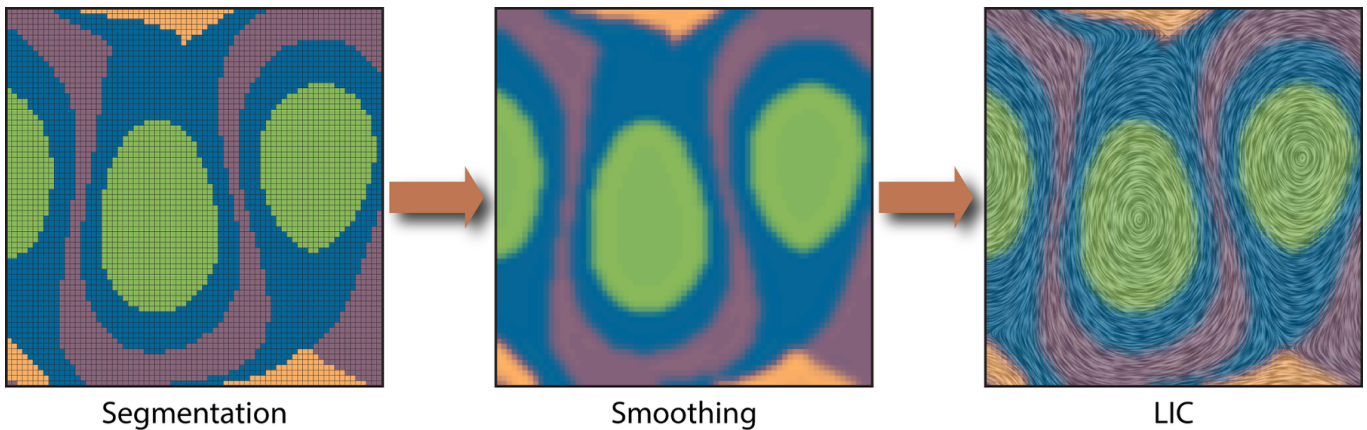


Fig. 5. Visualization pipeline.

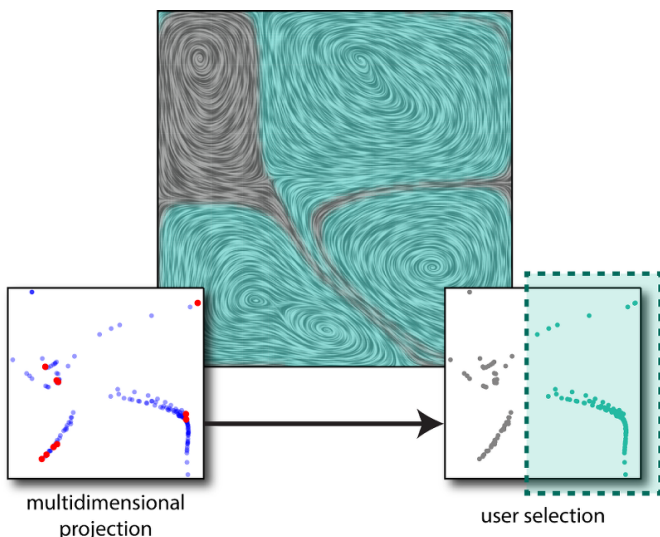


Fig. 4. Exploring structures in the vector field through user interaction in visual space (bottom).

the user with exploratory tools towards vector field interactive analysis.

As we expect that streamlines with similar shape will give rise to similar feature vectors, that is, their embedded counterparts are close to each other in the attribute space, the multidimensional projection should generate layouts where similar streamlines will be grouped together, allowing the user to interact with the layout towards exploring and visually mining structures of interest in the vector field.

Figure 4 shows a vector field where the highlighted regions correspond to the group of streamlines selected by the user. This example clearly shows the powerfulness of using multidimensional projection as an interactive tool in our context.

C. Clustering

The last step of our algorithm is to group the streamlines according to their similarity. This is accomplished by clus-

tering the projected feature vectors in the two-dimensional visual space. Clusters can be defined interactively by the user or automatically by applying a clustering algorithm, such as k-means. In our framework, the visual space is clustered by Ward’s minimum-variance method [14], a hierarchical clustering mechanism.

In order to highlight the regions in G containing similar streamlines, the information about the clusters is transferred back to G . This is performed in a straightforward way using the information of intersection (stored in the feature extraction step) between the cells $c_{ij} \in G$ and the streamlines. Since each streamline has a label that identifies which cluster it belongs to, a voting process is used in each cell c_{ij} and bind to c_{ij} the most frequent label of the streamlines that intersect c_{ij} .

III. RESULTS AND COMPARISONS

To generate the examples presented in this section we used a time step $\delta_t = 0.05$ for streamline integration and frequency histograms with 10 bins for each attribute in all datasets. As discussed in the last section, the features used to generate the feature vectors are: curvature, magnitude of vorticity, magnitude and divergence of velocity. In the following experiments, our main goal is to segment the regions in the vector field with different features, such as rotational, curvilinear, laminar, turbulent flow, etc.

The segmentation process produces a labeling of the grid cells, where cells with similar flow pattern have the same label (color code). Figure 1 shows the effectiveness of our technique to segment the von Kármán vortices (blue regions) and the flow structures around a cylinder. Our framework assigns distinct colors to represent distinct clusters and applying the LIC (line integral convolution) method [15] for visualization. In order to improve the quality of the visualization, we smooth the segmentation result in the grid using Garcia’s technique [16] before applying LIC. Figure 5 shows the visualization pipeline used to generate the results presented in this paper.

Next experiment, we take advantage of the user interaction enabled by LAMP. Figure 6 presents the vector field segmentation before and after the user driven control points (red

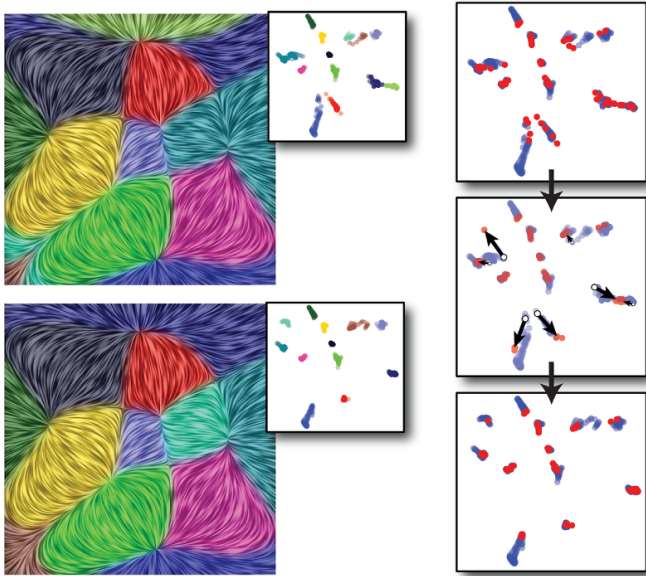


Fig. 6. User interaction with the control points in visual space (right) to improve the segmentation.

points) arrangements. Notice that the segmentation changes substantially after repositioning the control points in the visual space, illustrating the flexibility enabled by our framework to interactively intervene in the segmentation process resulting in well defined regions delimited by the vector field singularities (sources and sinks).

In order to attest the quality of the segmentation produced by our method, we compare it against the method proposed by Daniels et al. [9] which also makes use of feature vectors extracted from properties computed in a neighborhood of each grid cell using k -nearest neighbors (KNN) to visualize a vector field. More precisely, we computed the curvature, divergence, magnitude of velocity, and the intensity and direction of rotation of the vectors in each grid cell, concatenating those properties so as to generate feature vector in a high-dimensional space. Figure 7 left shows the resulting segmentation with the features extracted as proposed by Daniels. Notice that

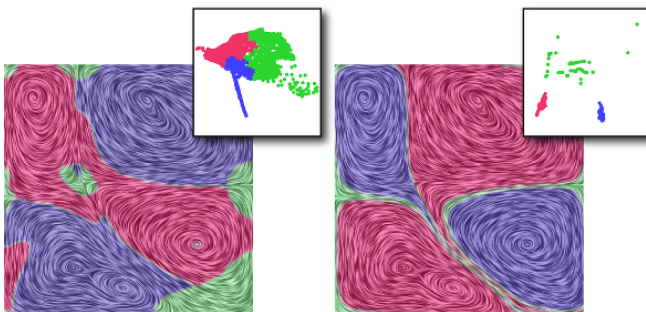


Fig. 7. Segmentation of the vector field from a smoke simulation. The left image shows the result of Daniels' method [9] using KNN with $k = 30$ for the samples. The right image shows the result of our method. Clusters of the points in visual space are shown on top-right.

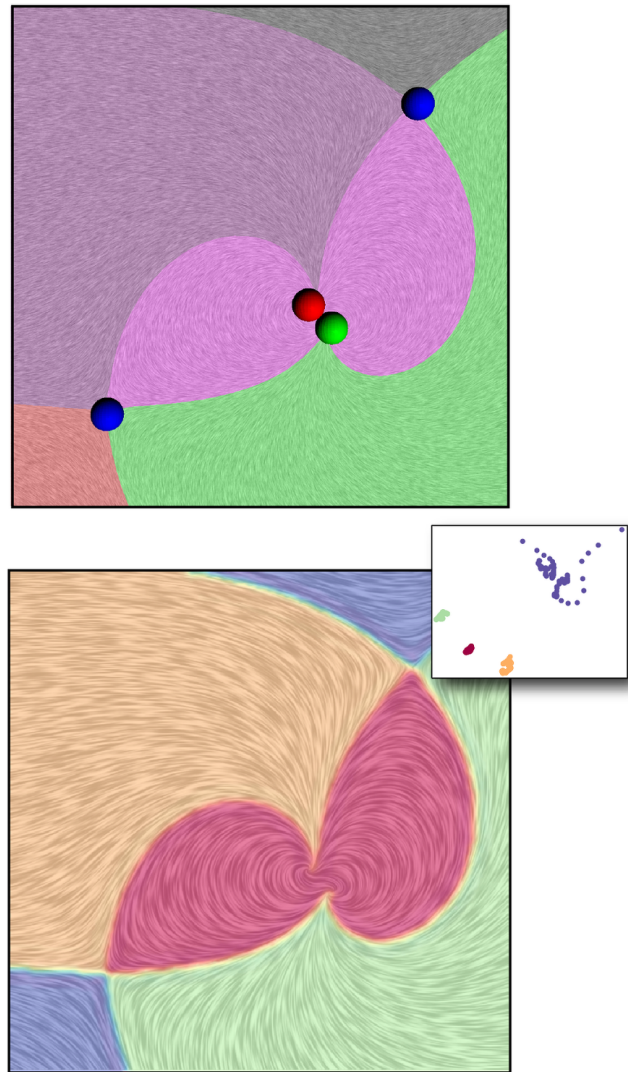


Fig. 8. A comparison between Edge Maps [5] (top) and our method (bottom).

the quality of the segmentation is much worse than the one produced by our methodology, as depicted in the right in Figure 7. It is important to say that, in his original paper, Daniels used a larger number of features, but we intend to show here that, in contrast to our approach, Daniels' method does not work properly with a reduced number of features.

Figure 8 compares our approach against the Edge Maps method [5] using a synthetic vector field. Besides Edge Maps relies on topological properties of the vector field using their critical points (colored spheres) to perform the segmentation and it is considered the state-of-art in the context of vector field segmentation, our approach is easy to implement and it compares positively with Edge Maps.

Our framework can be easily extended to 3D discrete vector fields. The visual metaphor adopted combines rendered streamlines of each cluster to provide a more informative visualization and a surface to highlight the transition zone between distinct clusters. The surface is computed using the

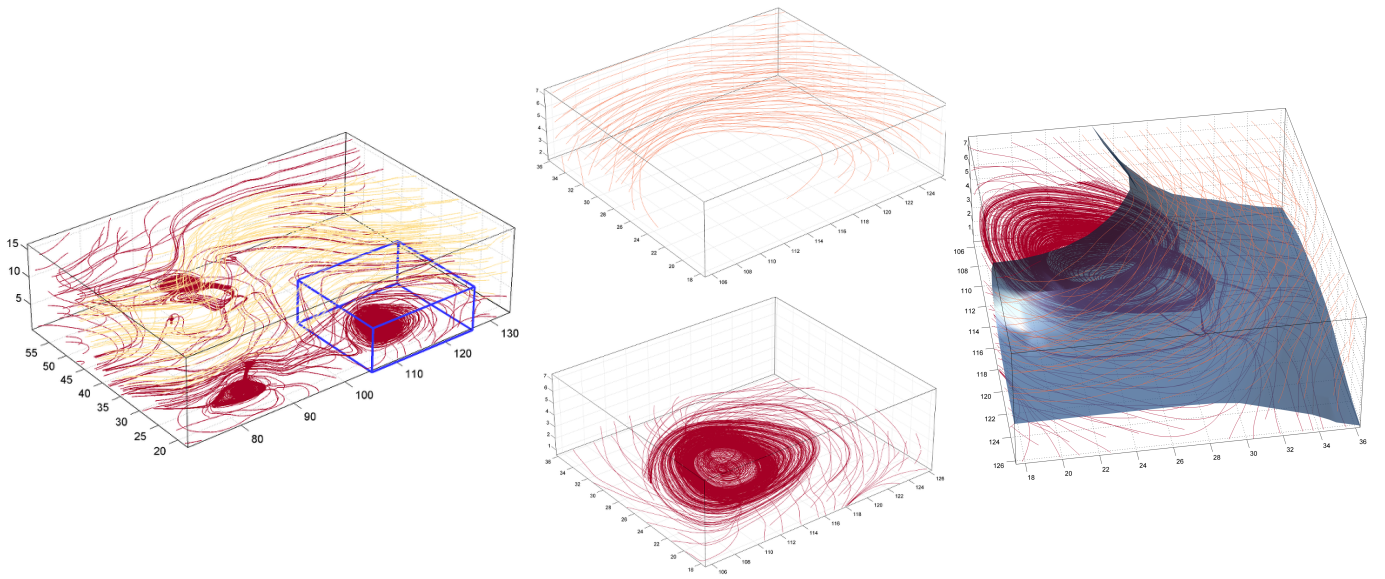


Fig. 9. Segmentation of a 3D discrete vector field. From left to right, segmentation in two clusters of the entire dataset, visualization of a sub-region (blue) and its surface of the transition zone between the clusters.

TABLE I
STATISTICS AND COMPUTATIONAL TIMINGS (IN SECONDS).

| Fig. | grid size | # streamlines | time |
|--------|-------------------------|---------------|-------|
| Fig. 1 | 197×436 | 1117 | 44.30 |
| Fig. 6 | 120×120 | 961 | 12.58 |
| Fig. 7 | 189×189 | 247 | 11.59 |
| Fig. 8 | 175×175 | 702 | 11.98 |
| Fig. 9 | $16 \times 14 \times 8$ | 141 | 21.89 |

Marching Cubes algorithm in the scalar field generated by the cluster labels and smoothed by Garcia’s method [16]. Figure 9 shows the segmentation using two clusters of a 3D vector field that represents air currents over North America (dataset from the MATLAB data library). In 3D case, we use an additional information of torsion as described by McLoughlin et al. [11].

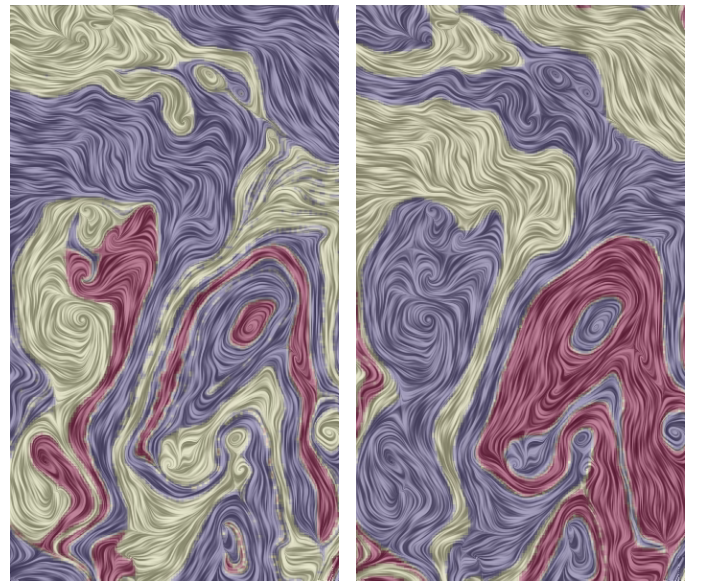
Computational performance is presented in Table I. Timings were measured in a 2.53 GHz Intel Core 2 Duo with 2GB RAM. MATLAB was used to implement the proposed framework.

IV. DISCUSSION AND LIMITATIONS

Our framework can handle a large variety of datasets. It is flexible as it permits the user to include the user knowledge to improve the segmentation results. We use streamlines as a sampling mechanism to generate feature vectors, which is one of the main contributions of this work. Moreover, other physical attributes, such as density and pressure could also be used, when available, without changing the proposed pipeline.

The comparisons show that our framework can capture flow structures quite nicely, generating results similar to highly accurate topological methods. Our method is more robust than the previous methods based on feature vectors to perform vector field segmentation.

The main limitation of our method is the lack of theoretical guarantees of the segmentation and in some cases artifacts can



(a) Noisy PIV data.

(b) Smoothed PIV data.

Fig. 10. Segmentation of PIV data from PIVsuite [17].

be produced generating not well defined clusters, which arise usually in noisy data.

Fig. 10 shows our method facing a noisy vector field from *particle image velocimetry* (PIV) [18] data set. Spurious results may be obtained in noisy vector fields (Fig. 10.a) and a possible solution consists in applying a vector field denoising process (Fig. 10.b). Notice that the features were able to discriminate the turbulent from the laminar region after smoothing step.

V. CONCLUDING REMARKS

In this paper we proposed a new method to segment and reveal related regions on vector fields with support to user interaction. Despite its simplicity, our method succeeds in many scenarios from artificial to “real world” collected data. In our tests, the proposed method achieved similar results as a state-of-the-art technique. Our technique is automatically extended to three-dimensional problems.

As future works tracking of features over time is an interesting topic and the parallelism of the sampling process should be used to speed-up gain. Another interesting topic for future research is the use of topological information to improve the multidimensional projection.

ACKNOWLEDGMENT

This research is supported by FAPESP (#2011/22749-8, #2013/07375-0, #2014/09546-9), CNPq (#305796/2013-5, #302643/2013-3) and CAPES (DS-7486558/M, PROEX). We would like to thank the anonymous reviewers for their valuable comments and suggestions.

REFERENCES

- [1] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramée, and H. Doleisch, “Feature extraction and visualization of flow fields,” *Eurographics 2002 STAR*, pp. 69–100, 2002.
- [2] K. Mahrous, J. Bennett, G. Scheuermann, B. Hamann, and K. I. Joy, “Topological segmentation in three-dimensional vector fields,” *IEEE Trans. Vis. Comput. Graph.*, vol. 10, no. 2, pp. 198–205, 2004.
- [3] R. Laramée, H. Hauser, L. Zhao, and F. Post, “Topology-based flow visualization, the state of the art,” in *Topology-based Methods in Visualization*, ser. Mathematics and Visualization. Springer, 2007, pp. 1–19.
- [4] H. Bhatia, S. Jadhav, V. Pascucci, G. Chen, J. A. Levine, L. G. Nonato, and P.-T. Bremer, “Edge maps: Representing flow with bounded error,” in *IEEE PacificVis 2011*, 2011, pp. 75–82.
- [5] —, “Flow visualization with quantified spatial and temporal errors using edge maps,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 9, pp. 1383–1396, 2012.
- [6] R. Nascimento, J. Paixão, H. Lopes, and T. Lewiner, “Topology aware vector field denoising,” in *Sibgrapi 2010 (XXIII Conference on Graphics, Patterns and Images)*, 2010, pp. 103–109.
- [7] H. Li, W. Chen, and I.-F. Shen, “Segmentation of discrete vector fields,” *IEEE Trans. Vis. Comput. Graph.*, vol. 12, no. 3, pp. 289–300, 2006.
- [8] A. Kuhn, D. J. Lehmann, R. Gaststeiger, M. Neugebauer, B. Preim, and H. Theisel, “A clustering-based visualization technique to emphasize meaningful regions of vector fields,” in *VMV 2011*, 2011, pp. 191–198.
- [9] J. Daniels II, E. Anderson, L. Nonato, and C. Silva, “Interactive vector field feature identification,” *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1560–1568, 2010.
- [10] C. Rossl and H. Theisel, “Streamline embedding for 3D vector field exploration,” *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 3, pp. 407–420, 2012.
- [11] T. McLoughlin, M. W. Jones, R. S. Laramée, R. Malki, I. Masters, and C. D. Hansen, “Similarity measures for enhancing interactive streamline seeding,” *IEEE Trans. Vis. Comput. Graph.*, vol. 99, pp. 1–1, 2012.
- [12] P. Joia, D. Coimbra, J. A. Cuminato, F. V. Paulovich, and L. G. Nonato, “Local affine multidimensional projection,” *IEEE Trans. Vis. Comput. Graph.*, vol. 17, no. 12, pp. 2563–2571, 2011.
- [13] E. Tejada, R. Minghim, and L. G. Nonato, “On improved projection techniques to support visual exploration of multi-dimensional data sets,” *Information Visualization*, vol. 2, no. 4, pp. 218–231, 2003.
- [14] E. Rasmussen, “Clustering algorithms,” *Information Retrieval: data structures and algorithms*, pp. 419–442, 1992.
- [15] B. Cabral and L. C. Leedom, “Imaging vector fields using line integral convolution,” in *SIGGRAPH’93*, 1993, pp. 263–270.
- [16] D. Garcia, “Robust smoothing of gridded data in one and higher dimensions with missing values,” *Comput. Stat. Data An.*, vol. 54, no. 4, pp. 1167–1178, 2010.
- [17] J. Vejrazka, “PIVsuite,” Institute of Chemical Process Fundamentals, Prague, Czech Republic, Tech. Rep., 2014, <http://www.mathworks.com/matlabcentral/fileexchange/45028-pivsuite>.
- [18] R. J. Adrian and J. Westerweel, *Particle Image Velocimetry*. Cambridge University Press, 2011.