

A User-Friendly Interactive Framework for Unsteady Fluid Flow Segmentation and Visualization

Danilo Motta · Wallace Casaca · Paulo Pagliosa · Afonso Paiva

Received: date / Accepted: date

Abstract While vector fields are essential to simulate a large amount of natural phenomena, the difficulty to identify patterns and predict behaviors makes the visual segmentation in simulations an attractive and powerful tool. In this paper, we present a novel user-steered segmentation framework to cope with steady as well as unsteady vector fields on fluid flow simulations. Given a discrete vector field, our approach extracts multi-valued features from the field by exploiting its streamline structures so that these features are mapped to a visual space through a multidimensional projection technique. From an easy-to-handle interface, the user can interact with the projected data so as to partition and explore the most relevant vector features in a guidance frame of the simulation. Besides navigating and visually mining structures of interest, the interactivity with the projected data also allows the user to progressively enhance the segmentation result according to his insights. Finally, to successfully deal with unsteady simulations, the segments previously annotated by the user are used as a training set for a Support Vector Machine (SVM) approach that classifies the remaining frames in the flow. We attest the effectiveness and versatility of our methodology throughout a set of classical physical-inspired applications on fluid flow simulations as depicted in the experiment results section.

Keywords Flow segmentation · time-varying visualization · vector field · interactive tools · machine learning

Danilo Motta and Afonso Paiva
Universidade de São Paulo (USP), ICMC, São Carlos, Brazil
E-mail: daniloam@icmc.usp.br E-mail: apneto@icmc.usp.br

Wallace Casaca
Universidade Estadual Paulista (UNESP), Rosana, Brazil
E-mail: wallace.coc@gmail.com

Paulo Pagliosa
Universidade Federal de Mato Grosso do Sul (UFMS), FACOM, Campo Grande, Brazil
E-mail: pagliosa@facom.ufms.br

1 Introduction

Segmenting vector fields is of key importance in the context of scientific illustration and visualization. The fluid mechanics community, for example, faces the problem when simulating the motion of a fluid, to capture different structures of the flow such as vortices, saddle points and skins [17]. Aiming at segmenting vector fields that commonly appear in computational simulations, three major categories have gained a great amount of attention in the last decade: topology-based, geometric-based, and feature space-based.

Methods that purely rely on topology extract critical points from the vector field so that the topological structures of these points are conveniently handled and classified to cluster different regions of the field [13, 11]. Despite their robustness in fragmenting arbitrary vector fields [2, 3], approaches based on topology are typically computationally expensive, being also intricate to be coded in most of the cases. Geometry-based methods profit from the geometric attributes of the vector field such as curvatures, gradients, and streamlines where the vector flow is partitioned accordingly the similarity between geometric representatives of the output segments. Although intuitive and effective [6, 12, 10, 20, 21], geometry-based methods often suffer from the absence of a user interactive interface to adequately customize the segmentation. Finally, the last class of methods takes a specified set of features computed from the vector field so that regions of similar flow behavior are grouped and classified [4, 18, 15]. In fact, feature space-based algorithms usually perform well when dealing with steady vector fields (a static frame extracted from the flow), however, processing time-dependent sequence of frames (the unsteady case) is still a hurdle for the majority of these methods.

In this work we introduce a new technique for segmenting both steady and unsteady vector fields which combines streamlines, multidimensional projection, and supervised machine learning into an innovative and unified framework. In contrast to most existing algorithms, in particular the three basic groups mentioned above that formally tackle the segmentation task as a well-behaved steady-state problem, our approach enables to segment multiple frames in a time-varying vector field, a challenging and more general case due to the preservation of the temporal coherence between the frames. Moreover, in our formulation the user can conveniently drive the segmentation by freely interacting with projected data obtained from a feature space in a 2D visual interface, leading therefore to alternatives for customization issues. As we shall show, the clusters annotated by the user in a given frame of the vector flow are automatically propagated to the others by applying modern concepts of machine learning, preserving the user's choice while still avoiding unnecessary side processing in all the remaining frames of the simulation.

2 Pipeline Overview

Our supervised segmentation framework is composed by four main steps, namely, *feature extraction*, *multidimensional projection*, *clustering*, and *learning*, as presented in Figure 1. In the first step, streamlines are extracted from the vector field and embedded into a multi-attribute space whose dimensions correspond to differential and geometric properties of the vectors, being computed for each time-step (a frame) of the flow. The resulting high-dimensional data is projected to a 2D visual space so as to preserve neighborhood structures between the streamlines of the vector field as much as possible. Clustering is then performed in the visual space accordingly the prior subdivision of partitions by the viewer, thus highlighting the geometric patterns that s/he intends to visualize in the simulation. Finally, in the

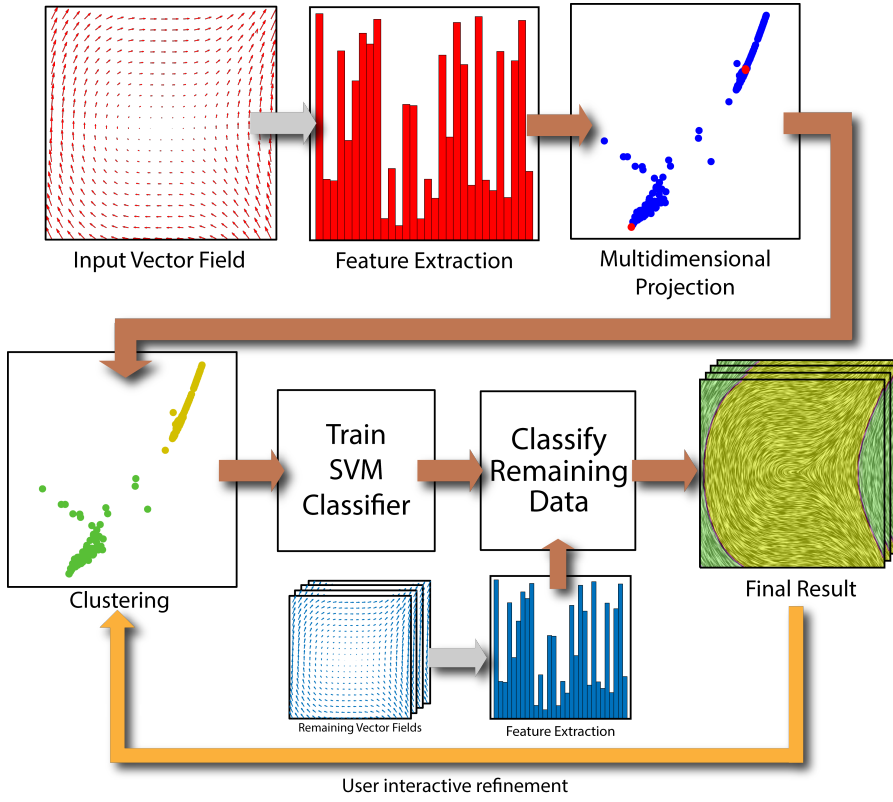


Fig. 1 Pipeline overview of our vector field segmentation framework.

last stage, we train a Support Vector Machine (SVM) mechanism to automatically propagate the traits previously specified by the user to the remaining frames of the vector field. Another interactive aspect of our framework is that the viewer can also assign new clusters to be exploited once the simulation is being run (see the bottom arrow in Figure 1).

As a boosting step for our approach, we run the feature extraction apparatus described in [15] aiming at producing a set of high-dimensional points for each frame of the simulation. This gives rise to our multidimensional projection scheme, which is carried out by the LAMP algorithm [9]. In the following, we briefly discuss the technical aspects regarding the flow features used to construct our multidimensional attribute space.

3 Features Description and Quantization

Given a streamline S obtained as in [15], we extract a set of discriminative features representing different properties of the flow so that these collected attributes are quantized and prescribed as feature vectors into a high-dimensional space. These attributes are numerically computed from the vector field V by employing basic Finite Difference (FD) rules and then interpolated for each streamline point $s_i \in S$ through a regular bicubic (or tricubic, in the 3D case) interpolation scheme. So, let \mathbf{v} be the local velocity at a certain streamline point

$s_i \in S$. We then consider the following attributes to represent our collection of differential and geometric flow features:

1. *Magnitude of Velocity* ($\|\mathbf{v}\|$): determines the intensity of the resulting velocity in a certain point of the vector field V [19].
2. *Magnitude of Vorticity* ($\|\boldsymbol{\omega}\|$): measures how much a fluid portion locally rotates as it moves with its flow [19].
3. *Divergence* ($\nabla \cdot \mathbf{v}$): gauges how much a fluid portion is being transported away from the neighborhood of a given point located in the flow, in other words, it quantifies the decay rate of the density at this point [19].
4. *Curvature* (κ): can be intuitively understood as a measure of how sharply a curve is turning as it is traversed. In our implementation, we compute the curvature as suggested by McLoughlin et al. [14]:

$$\kappa = \frac{\|\mathbf{v} \times \mathbf{a}\|}{\|\mathbf{v}\|^3}, \quad (1)$$

where \mathbf{a} is the local acceleration, computed by $\mathbf{J} \cdot \mathbf{v}$, with $\mathbf{J}_{ij} = \partial v_i / \partial x_j$ being the coefficients of the Jacobian matrix.

5. *Torsion* (τ): accounts for the twisting of a particle embedded within a 3D vector field. So, we include τ as a valid attribute for 3D simulations, which is computed as follows [14]:

$$\tau = \frac{(\mathbf{v} \times \mathbf{a}) \cdot (\mathbf{J} \cdot \mathbf{a})}{\|\mathbf{v} \times \mathbf{a}\|^2}. \quad (2)$$

In order to assign a single feature vector to each streamline S , an attribute value A_i^k for each one of the above-mentioned features k is designed to represent the attributes of $s_i \in S$. These values are then distributed, by features, to form a frequency histogram of m bins. Finally, a composite histogram is created by concatenating the feature histograms into a $4m$ -dimensional attribute vector (see Figure 2), in summary: a streamline S is represented by a $4m$ -dimensional point in a feature space. In our experiments we take $m = 10$, which leads to high-dimensional spaces of 40 and 50 dimensions for 2D and 3D vector fields, respectively.

Next, we describe the key aspects of the high-dimensional mapping to the visual plane to properly evidence the good capabilities and flexibility of our approach to the task of segmenting and learning structures of interest in arbitrary dynamic vector field simulations.

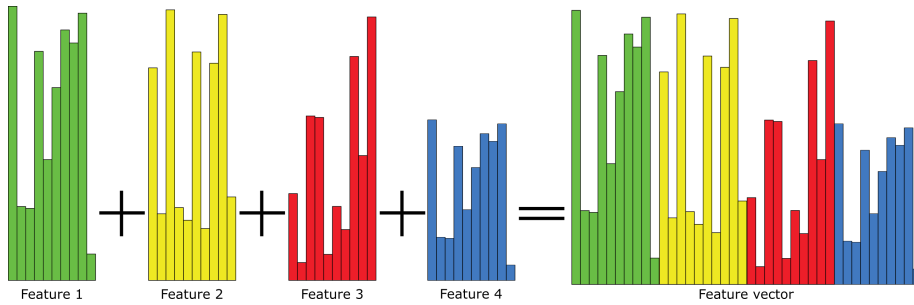


Fig. 2 Frequency histogram generated by concatenating the histograms derived from the above-mentioned features to compose a feature vector of 40 components in the high-dimensional attribute space.

4 Multidimensional Projection

In this step, we focus on mapping the high-dimensional instances obtained from the feature extraction process onto a two-dimensional visual space. For this purpose, we use the multidimensional projection technique called LAMP (*Local Affine Multidimensional Projection*) [9] to project the data as well as to assist the segmentation process, in an effort to discriminating patterns of interest into separated clusters. Besides preserving neighborhood structures efficiently during the projection step, the LAMP technique also enables a very flexible mechanism to interactively modify the projection according to the user instructions. Notice that this is an important trait in our context (which also covers time-varying vector fields), as the process of segmenting “frame-by-frame” the whole simulation tends to be laborious and very time-consuming. Further, by using this kind of interactive proposal, one could also avoid the preliminary task of segmenting completely a boosting simulation frame before propagating its segments to the other frames, which indeed leads to be more applicable and opportune for practical usages.

LAMP takes a set of control points to accomplish the mapping \mathbf{T} of a high-dimensional set of points \mathbf{X} to a two-dimensional visual space \mathbf{Y} . The control points are typically a small subset $\mathbf{X}_S \subset \mathbf{X}$ whose basic information contained in the interactive plane \mathbf{Y} is known a priori. The mapping \mathbf{T} is achieved by solving the following optimization problem:

$$\underset{\mathbf{T}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{T} - \mathbf{B}\|_F \quad \text{subject to} \quad \mathbf{T}^\top \mathbf{T} = \mathbf{I}, \quad (3)$$

where $\|\cdot\|_F$ stands for the Frobenious norm, \mathbf{I} is the identity matrix, and \mathbf{A} and \mathbf{B} are matrices whose rows correspond to the coordinates of each control point in the high-dimensional and visual space, respectively. The orthogonal constraint $\mathbf{T}^\top \mathbf{T} = \mathbf{I}$ enforces that the resulting transformation \mathbf{T} will behave like a rigid transformation, ensuring that the distances will be preserved as much as possible after the data projection. Equation (3) is efficiently solved through the Singular Value Decomposition (SVD) method as pointed out in [9], i.e.:

$$\mathbf{T} = \mathbf{U}\mathbf{V}^\top, \quad \text{with} \quad \mathbf{A}^\top \mathbf{B} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad (4)$$

where $[\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}] = \text{SVD}(\mathbf{A}^\top \mathbf{B})$. When a user interacts with the control points in the visual space, s/he also modifies the entries in the rows of matrix \mathbf{B} , thus adjusting \mathbf{T} to deal with the user intervention. It means that, technically speaking, when a control point is dragged in the visual space, the data in its neighborhood is also dragged together, which allows for controlling neighborhood structures directly from the visual space. Such a flexibility in handling neighborhood structures is exploited in our context, enabling to the user a robust exploratory mechanism towards vector field interactive analysis.

Once the high-dimensional instances have been projected, the mapped points are then used to support the next step of our framework as an effective and exploratory tool to interactively label and mining structures of interest in the vector field.

4.1 User Manipulation of Clusters

Considering the flexibility in operating distinct collections of data provided by Eq. (3), we take advance of this fact towards locating and visually mining clusters of streamlines within the vector field. More specifically, users are invited to actively participate in the segmentation process by organizing the data into desirable segments of streamlines. Given a reference frame F_j among the complete list of vector field layers F_i , $i = 1, 2, \dots, n$, the user selects in

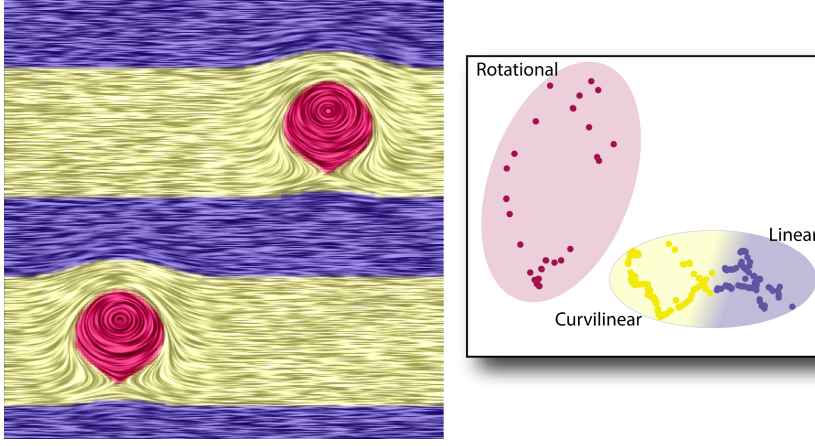


Fig. 3 Segmentation of a target frame in a time-dependent vector field. The clusters selected by the user are shown in the visual interactive layout (right), and the segmentation generated after user interaction (left).

the interactive layout the group of features that s/he would like to give prominence in F_j (see Figure 3). Since our framework is easy-to-handle and quite intuitive, its use can allow both scientists as well as non-familiar viewers to interact with the streamlines in a vector field, therefore facilitating the visual identification of important singularities in a computational simulation, for example, to locate sources and sinks. Furthermore, the identification of clusters is also facilitated during the visual navigation, as the clusters are naturally seen due to their geometric positions in the projection plane as previously illustrated in Figure 3.

Besides easy-to-operate and robust in terms of clustering similar streamline-coded points, our technique also enables an interactive mechanism which allows for repartitioning data by “dragging and drop” the projected control points. When a control point is moved in the interactive plane, the data in its neighborhood is moved together, thus allowing to control the neighborhood structures in the visual space. So, if the projection result is not satisfactory, the user can select badly clustered points, turning them into new control points that can be manually moved in the interactive layout to better refine clusters and, thereby, improve the streamlines classification. Figure 4 illustrates a user intervention case where control points are dragged to different clusters within the interactive layout.

4.2 Control Points Assignment

Let N be the number of streamline vectors in \mathbf{X} taken from a given guidance frame F_j , and n_S be the number of control points in the projection plane \mathbf{Y} . As suggested in [9], we set $n_S = \lceil \sqrt{N} \rceil$ as a prior estimation to pave the way for reaching a good level of accuracy in our opening multidimensional mapping. So, as a useful and predictive scheme to get the best candidates for the control points, we firstly subdivide the vector field domain into a d -dimensional regular grid \mathbf{G} containing $\lceil \sqrt[d]{n_S} \rceil$ cells in each axis, wherein d stands for the vector field dimension. For each cell $C \in \mathbf{G}$, let $\mathbf{F}_C = \{\phi_i\}_{i=1}^N$ be the set of feature vectors corresponding the streamlines which intersect C . The control point associated to the cell C is the feature vector $\phi_j \in \mathbf{F}_C$ which minimizes the following expression:

$$j = \underset{i}{\operatorname{argmin}} \|\phi_i - \mathbf{M}_C\|_2^2, \quad (5)$$

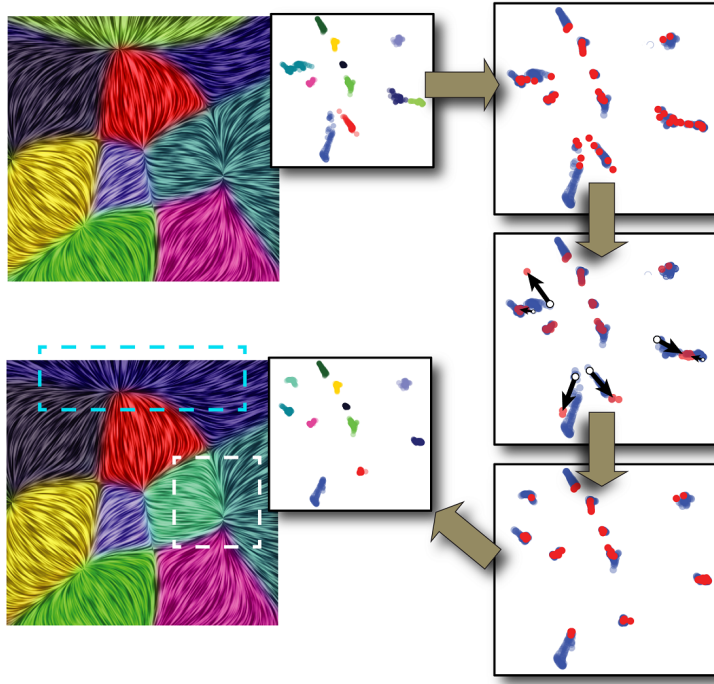


Fig. 4 Manipulation of control points by drag-and-drop interactions. The user can interact with the vector field partitions by re-positioning control points (in red) in the visual space (see the movements of the black arrows in the middle-right layout) in order to enhance the segmentation quality (see the dashed areas).

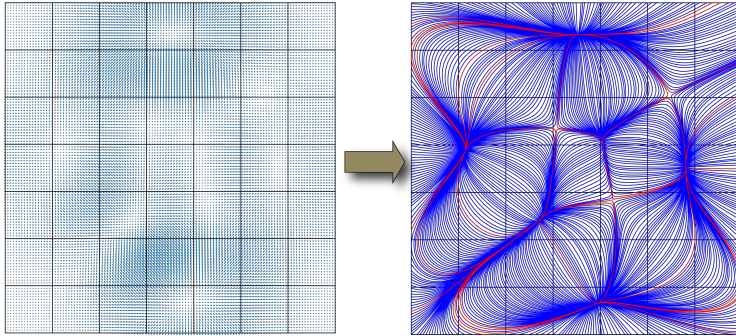


Fig. 5 Control points selection strategy. (Left) The vector field is subdivided into a regular grid. (Right) For each grid cell C , we compute the mode \mathbf{M}_C of all feature vectors whose corresponding streamlines (in blue) intersect C . The control point associated to C is the feature vector closer to \mathbf{M}_C (see the red streamline).

where the vector \mathbf{M}_C accounts for a statistical measure of centrality given by the mode among all feature vectors in \mathbf{F}_C .

This strategy enables a more refined selection of control points, embedding different behaviors across the vector field in a more equalized and representative manner. Moreover, it solves the problem of having distinct projections for the same dataset, as our procedure to select control points is totally deterministic. Figure 5 illustrates the employed scheme.

5 Learning Segmentation on Unsteady Vector Fields

Let $\tilde{\mathbf{X}}$ be the multidimensional feature space formed by all streamline-like points taken from the frames F_i over time, $i = 1, 2, \dots, n$. We start from the projected 2D points grouped as multiple clusters by the user in the guidance frame F_j managed as in the previous step of our framework. So, clusters as well as the high-dimensional data $\tilde{\mathbf{X}}$ are used to drive the streamlines classification in all time-dependent frames F_i , $i \neq j$. More precisely, we create a training set for a machine learning method which learns and sorts out the feature points of the frames accordingly their level of similarity with the features of F_j . To accomplish this task, we employ a multi-class version of the *Support Vector Machine* (SVM) [5] on $\tilde{\mathbf{X}}$.

We apply the SVM to construct a training set \mathbf{Q} which reflects the user-specified traits from the guidance frame F_j . This training set is used to generate a set of hyperplanes $\tilde{\mathbf{H}}$ in $\tilde{\mathbf{X}}$ so that $\tilde{\mathbf{H}}$ splits $\tilde{\mathbf{X}}$ into multiple disjoint subspaces. As a result, the sampled points from $\tilde{\mathbf{X}}$ are classified so that each streamline located at F_i , $i \neq j$, is labeled in terms of a cluster learned in F_j . In practice, to reduce the influence of outliers on $\tilde{\mathbf{X}}$, we also introduce in the classification process the lower and upper bounds of histograms encoded by the streamlines. Such a procedure is performed by imposing for each set of multi-featured frame points, the global average of the histogram bounds as they range all the unclassified frames.

The strength and versatility of the machine learning theory smooth the path to properly manage a large amount of frames while achieving multiple segmentations. Indeed, this learning scheme turns the segmentation task simpler and less laborious on unsteady vector fields, avoiding the exhaustive and individual supervision of the frames by the user. Furthermore, regions with similar streamlines in the vector field can be easily identified from the projected data in the visual space, pointing out where users should nail down their exploration in order to propagate to the other frames the patterns previously labeled in F_j .

Finally, another useful aspect of our approach is the availability for interactive reclassifications, as we can seen in Figure 6. First, the user defines a cluster setup in the visual space before applying the training and classification process. So, from the simulation being run, s/he rearranges the clusters as observed in the bottom row of Fig. 6 to create a new training set and, hence, also a novel time-varying segmentation that matches the user's purpose.

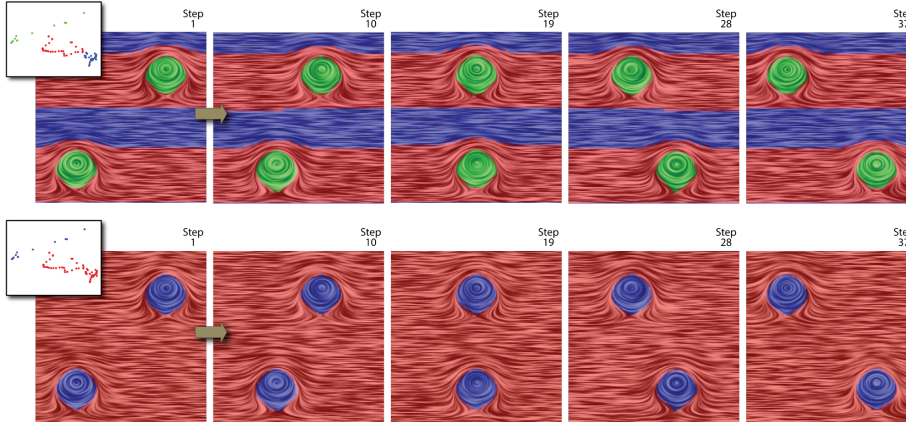


Fig. 6 Interactively retraining our framework. The user at first clusters the vector field, analyses the dynamic results (top) and, later, retrains our framework with a secondary cluster configuration to determine a new classifier and, therefore, also a novel time-varying segmentation (bottom).

6 Experiments

In order to demonstrate the robustness and flexibility of our approach to segment arbitrary vector fields, we compare it against two state-of-the-art visualization methods namely *Daniels' method* [4] and the *Edge maps tool* [3]. We also evaluate our framework in a set of challenging cases comprising time-varying vector fields (the unsteady case), as the ones conducted by classical simulations of Computational Fluid Dynamics (CFD) such as *Rayleigh-Taylor effect* and *von Kármán instabilities*. So, we focus on properly classifying the vector fields into multiple clusters so that they can be seen by the user as a junction of different flow patterns (e.g., rotational, curvilinear, laminar, turbulent flow, etc).

To computationally create our numerical simulations, we employ the well-known Gerris solver [16]. Considering the unsteady set of experiments, we take a time step $\delta_t = 0.05$ for the streamline integration and we assign 10 for the number of bins of the frequency histograms (see Sec. 3). Finally, as described before, we make use of the following geometric features to create multi-valued vectors to be used in our classification scheme: *curvature*, *divergence*, *magnitude of velocity/vorticity*, and *torsion* (for 3D vector fields, only).

6.1 The Steady Case

We start our tests comparing the proposed approach versus Daniels et al. method [4] when both are applied to segment a smoke flow vector field. Figure 7 (left) depicts the segmentation of the smoke flow with the features extracted as proposed in [4], while Figure 7 (right) shows the output produced by our technique. Notice from the rightmost layout displayed in Figure 7 that our framework performs well when compared to [4], as the projected data was better spread in the visual space from our technique, enabling the user to quickly identify natural clusters during the visual inspection. Another aspect is that the proposed framework leads to more structured clusters in the vector field as well as the lack of artifacts in the segmented regions (e.g. see in the leftmost image the pink gap inside the greenish partition).

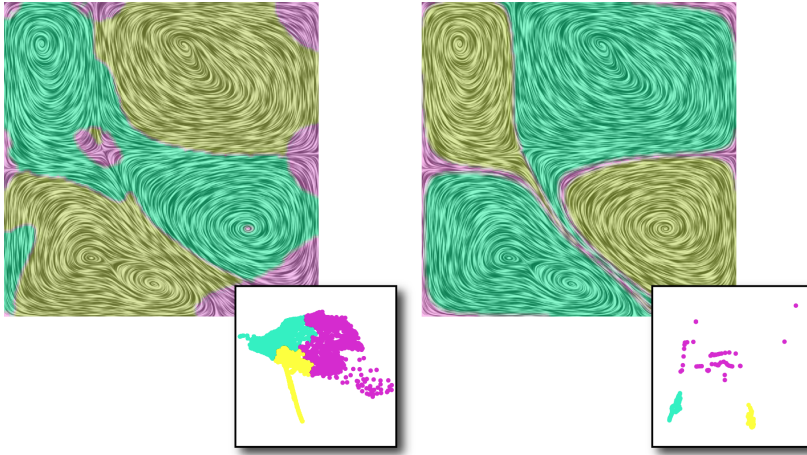


Fig. 7 Segmentation of the vector field from a smoke simulation. The left image shows the result of Daniels' method [4] using KNN (k -nearest neighbors) with $k = 30$ for the samples. The right image brings our result. The projections are also portrayed in the bottom-right layouts of the vector fields.

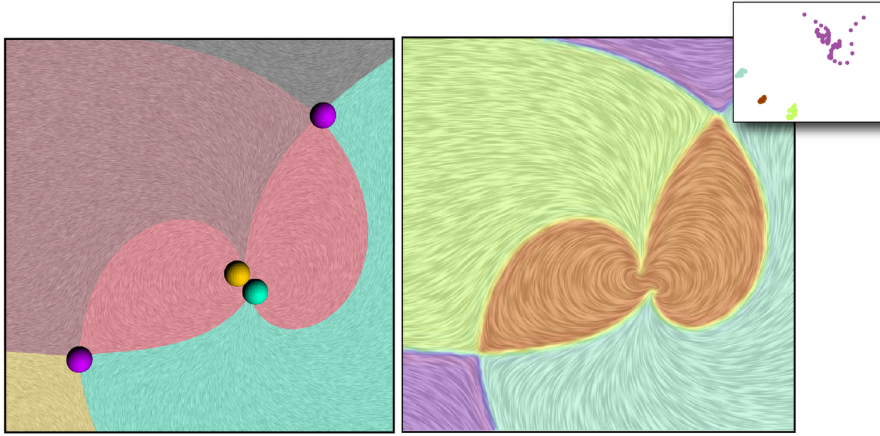


Fig. 8 Comparison between the Edge Maps method (left) and our methodology (right). Critical points (colored spheres) are depicted in the Edge Maps' result while both visual layout and segmentation are portrayed in the rightmost image for our method.

Figure 8 demonstrates the good performance of the proposed methodology when evaluated against the well-established *Edge Maps method* [3], a powerful state-of-the-art technique which takes advantage of topological properties of the vector field to accomplish the clustering. One can verify from the results depicted in Figure 8 that our approach originates quite similar partitions to the ones coming from the Edge Maps, attesting the positive performance of the designed technique even though against a robust (although a labor-intensive tool to be properly coded and tuned) visualization tool such as Edge Maps method.

Another important characteristic to be emphasized is that, in contrast with the visualizations typically produced by the Edge Maps, our method holds the capacity of interactive exploration, providing to the viewers a simple and versatile tool to explore and inspect regions accordingly their own interest, besides the benefits acquired from their insights and prior knowledge when interacting with the vector field. Another issue is the computational cost of the EM algorithm, which is considered higher than our method.

6.2 The Unsteady Case

We now concentrate our experimental analysis towards dealing with the harder case of unsteady fluid flow simulations, that is, when the flow varies as the time increases. For this purpose, we consider three simulations as reported below.

Figure 9 presents the resulting segmentation for a vector field obtained from the von Kármán vortex street simulation. Notice from the representative frames in Figure 9 that our approach is able to accurately detect the swirling-like vortices caused by the subdivision of the fluid flow around the cylindrical object. Moreover, our technique also accurately captures the vortices even to the more drastic case when they totally detach from the cylinder (see the transitions from steps 273 to 276, and 324 to 344)¹.

¹ Check the complete video: <http://icmc.usp.br/e/13837>

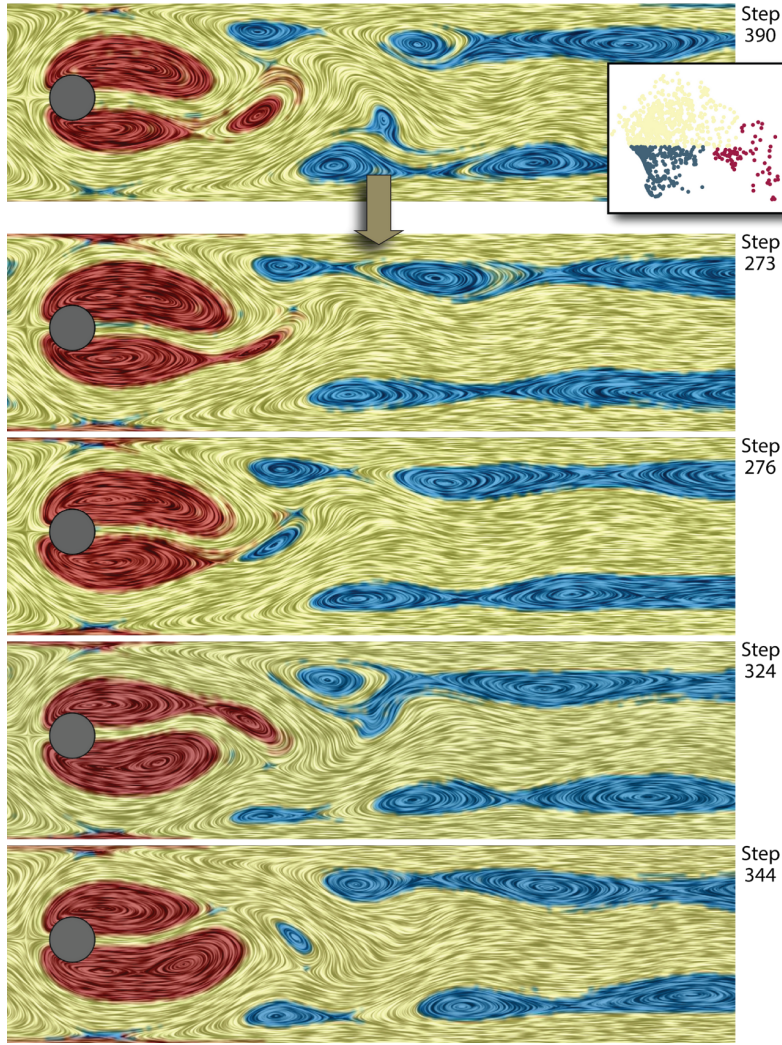


Fig. 9 Unsteady vector field segmentation of a von Kármán vortex street generated by a cylindrical object (brown). (Top) User-steered classification in an arbitrary frame allows the method to propagate the learned prior segmentation to the remaining layers in the simulation. Note that the higher velocity vortices (red) are dismembered from the vortices with lower velocity (cyan).

Rayleigh-Taylor (RT) instability occurs between two fluids of different densities when the lighter fluid pushes and wraps the second one at interface into vortices. Figure 10 elucidates this phenomenon wherein a given frame of the RT simulation (see the step 35) is used as a guidance segmentation by our training process so that the other frames of the vector field are classified and labeled through our machine learning model. As illustrated in Figure 10 (top row), our approach precisely captures the symmetric vortices along the simulation frames. In fact, this symmetry can also be verified on the unstable behavior of the fluids in the corresponding bottom row layers of Figure 10¹.

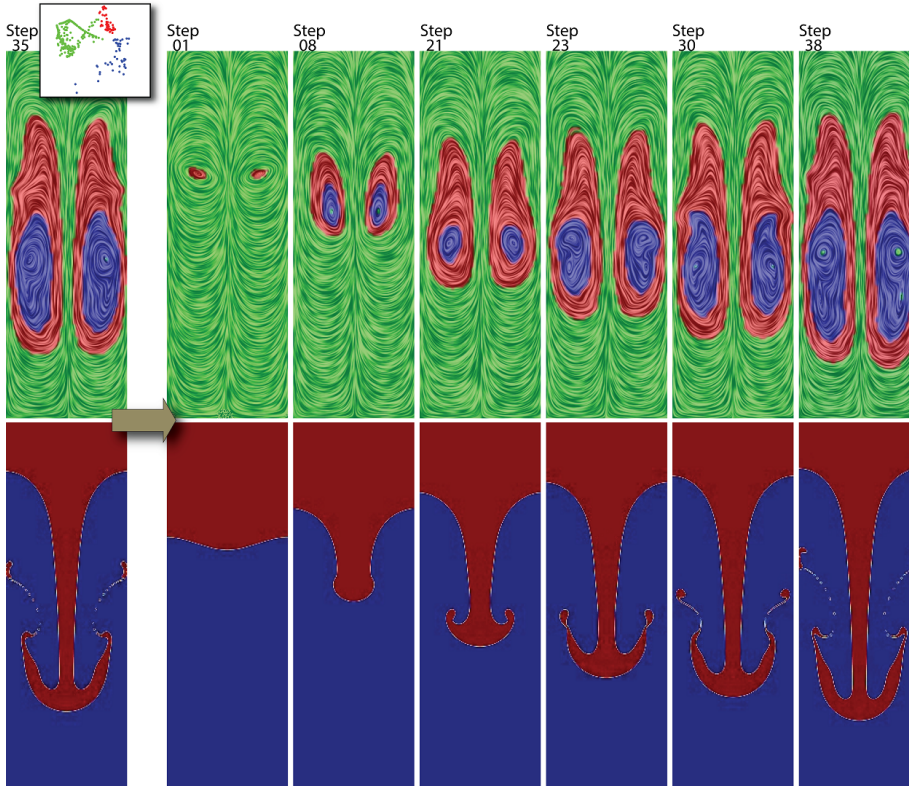


Fig. 10 Projected and user segmented data (topmost-left); clusters propagated along the flow steps (top left-to-right). On the bottom, the interface between a heavier (red) and lighter (blue) fluid which simulates the well-known Rayleigh-Taylor instability.

Our method can also be easily extended to three-dimensional discrete vector fields. The visual metaphor adopted in this case combines rendered streamlines for each cluster to provide a more informative visualization of 3D simulations, and a surface to better highlight the transition zone between distinct partitions. The surface is generated using the Marching Cubes algorithm in a scalar field built from the labels of the clusters smoothed by Garcia’s method [8]. Figure 11 shows the segmentation of two clusters in a 3D flow obtained from a resampled version of the Von Funck et al’s dataset [7] for smoke visualization. Note that our method reaches a good level of partitioning even to the three-dimensional simulation case of the von Karman street phenomenon, as the characteristic vortices are subdivided into two different classes of flows: laminar/curvilinear and turbulent regions.

Computational performance of the performed simulations is summarized in Table 1. We run all the reported experiments on an Intel Xeon E5345 with eight 2.3 GHz cores and 16GB RAM, being the simulation steps processed among these eight cores. Another important aspect regards computational issues of our approach is that the streamline integration, interpolation and data processing can also be run in parallel for each step of our simulation so that the timings described in Table 1 can be further improved in the future.

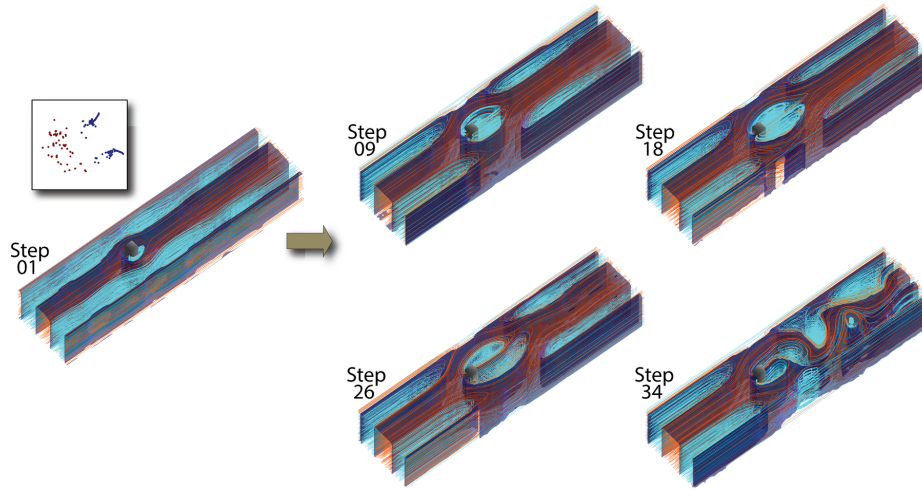


Fig. 11 Segmentation of a 3D flow around a parallelepiped. From left-to-right: reference and captured frames along the time, respectively.

Table 1 Steady and Unsteady vector fields characteristics and timings (in sec).

Experiment	# Frames	Grid size	# Streamlines	Time (sec)	Category
Figure 3	1	161×161	251	12.03	Steady
Figure 4	1	90×90	961	8.66	Steady
Figure 9	400	400×100	1639	7046	Unsteady
Figure 10	38	150×50	1372	318	Unsteady
Figure 11	50	$32 \times 96 \times 24$	2204	8615	Unsteady

7 Conclusion

In this paper we address the problem of segmenting steady as well as unsteady fluid flow simulations as an interactive framework that unifies multidimensional projection, machine learning concepts and the user capability in grouping data into different clusters of interest. Despite its simplicity, flexibility and robustness in segmenting vector fields, our approach can succeed in many scenarios, ranging from artificial to “real world” collected data, as verified by the experimental results. Moreover, the proposed methodology achieves very pleasant results when compared to existing representative techniques and also providing useful traits as interactivity flexibility and easiness of use. In summary, all these characteristics render the proposed framework a very attractive interactive tool in the context of fluid flow visualization, specially to handle the complex case of unsteady simulations.

As future work, we intend to incorporate new features extracted from the vector field into our framework, such as the energy of the flow provided by the POD (*Proper Orthogonal Decomposition*) [1] modes. Another direction is to port our method to GPU architecture since the classification of unsteady vector fields can also be accomplished in parallel.

Acknowledgment

We would like to thank Harsh Bhatia to kindly provide the results with Edge Maps [3]. This research has been supported by FAPESP (São Paulo Research Foundation: grants #2013/07375-0; #2014/16857-0), and CAPES (PDSE #133553/2016-01).

References

1. Berkooz, G., Holmes, P., Lumley, J.L.: The proper orthogonal decomposition in the analysis of turbulent flows. *Annual Rev. Fluid Mech* pp. 539–575 (1993)
2. Bhatia, H., Jadhav, S., Pascucci, V., Chen, G., Levine, J.A., Nonato, L.G., Bremer, P.T.: Edge maps: Representing flow with bounded error. In: *IEEE PacificVis 2011*, pp. 75–82 (2011)
3. Bhatia, H., Jadhav, S., Pascucci, V., Chen, G., Levine, J.A., Nonato, L.G., Bremer, P.T.: Flow visualization with quantified spatial and temporal errors using edge maps. *IEEE Transactions on Visualization and Computer Graphics* **18**(9), 1383–1396 (2012)
4. Daniels II, J., Anderson, E., Nonato, L., Silva, C.: Interactive vector field feature identification. *IEEE Transactions on Visualization and Computer Graphics* **16**(6), 1560–1568 (2010)
5. Dietterich, T.G., Bakiri, G.: Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research* pp. 263–286 (1995)
6. Ferreira, N., Klosowski, J., Scheidegger, C., Silva, C.: Vector field k-means: clustering trajectories by fitting multiple vector fields. *Computer Graphics Forum* **32**, 201–210 (2013)
7. von Funck, W., Weinkauff, T., Theisel, H., Seidel, H.P.: Smoke surfaces: An interactive flow visualization technique inspired by real-world flow experiments. *IEEE Transactions on Visualization and Computer Graphics* **14**(6), 1396–1403 (2008). URL <http://tinoweinkauff.net/>
8. Garcia, D.: Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics & Data Analysis* **54**(4), 1167–1178 (2010)
9. Joia, P., Coimbra, D., Cuminato, J.A., Paulovich, F.V., Nonato, L.G.: Local affine multidimensional projection. *IEEE Transactions on Visualization and Computer Graphics* **17**(12), 2563–2571 (2011)
10. Kuhn, A., Lehmann, D.J., Gaststeiger, R., Neugebauer, M., Preim, B., Theisel, H.: A clustering-based visualization technique to emphasize meaningful regions of vector fields. In: *Vision, Modeling, and Visualization* (2011), pp. 191–198 (2011)
11. Laramée, R., Hauser, H., Zhao, L., Post, F.: Topology-based flow visualization, the state of the art. In: *Topology-based Methods in Visualization, Mathematics and Visualization*, pp. 1–19. Springer (2007)
12. Li, H., Chen, W., Shen, I.F.: Segmentation of discrete vector fields. *IEEE Transactions on Visualization and Computer Graphics* **12**(3), 289–300 (2006)
13. Mahrous, K., Bennett, J., Scheuermann, G., Hamann, B., Joy, K.I.: Topological segmentation in three-dimensional vector fields. *IEEE Transactions on Visualization and Computer Graphics* **10**(2), 198–205 (2004)
14. McLoughlin, T., Jones, M.W., Laramée, R.S., Malki, R., Masters, I., Hansen, C.D.: Similarity measures for enhancing interactive streamline seeding. *IEEE Transactions on Visualization and Computer Graphics* **99**, 1–1 (2012)
15. Motta, D., Oliveira, M., Pagliosa, P., Nonato, L.G., Paiva, A.: Exploratory segmentation of vector fields using multidimensional projection. In: *Sibgrapi 2015 (28th Conference on Graphics, Patterns and Images)*, pp. 250–256 (2015)
16. Popinet, S.: Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics* **190**(2), 572–600 (2003)
17. Post, F.H., Vrolijk, B., Hauser, H., Laramée, R.S., Doleisch, H.: Feature extraction and visualization of flow fields. *Eurographics 2002 STAR* pp. 69–100 (2002)
18. Rossli, C., Theisel, H.: Streamline embedding for 3D vector field exploration. *IEEE Transactions on Visualization and Computer Graphics* **18**(3), 407–420 (2012)
19. Sternberg, S.Z., Loomis, L.H.: *Advanced Calculus: Revised Edition*. World Scientific Publishing Company (2014)
20. Zhang, L., Chen, G., Laramée, R.S., Thompson, D., Sescu, A.: Flow visualization based on a derived rotation field. *Electronic Imaging* **2016**(1), 1–10 (2016)
21. Zhang, L., Laramée, R.S., Thompson, D., Sescu, A., Chen, G.: An integral curve attribute based flow segmentation. *Journal of Visualization* **19**(3), 423–436 (2016)