

## An Image-Space Morse Decomposition of Vector Fields\*

Guoning Chen<sup>†</sup> and Shuyu Xu<sup>†</sup>

Department of Computer Science  
University of Houston  
Houston, TX, 77204, USA  
<http://www.cs.uh.edu>

Technical Report Number UH-CS-14-03

July 28, 2014

**Keywords:** Vector fields, Morse decomposition, image-space, stability analysis.

### Abstract

Morse decompositions have been proposed to compute and represent the topological structure of steady vector fields. Compared to the conventional differential topology, Morse decomposition and the resulting Morse Connection Graph (MCG) is numerically stable. However, the granularity of the original Morse decomposition is constrained by the resolution of the underlying spatial discretization, which typically results in non-smooth representation. In this work, an Image-Space Morse decomposition (ISMD) framework is proposed to address this issue. Compared to the original method, ISMD first projects the original vector field onto an image plane, then computes the Morse decomposition based on the projected field with pixels as the smallest elements. Thus, pixel-level accuracy can be achieved. This ISMD framework has been applied to a number of synthetic and real-world steady vector fields to demonstrate its utility. The performance of the ISMD is carefully studied and reported. Finally, with ISMD an ensemble Morse decomposition can be studied and visualized, which is shown useful for visualizing the stability of the Morse sets with respect to the error introduced in the numerical computation and the perturbation to the input vector fields.



\*This work was supported by the National Science Foundation, IIS-1352722.

<sup>†</sup>Guoning Chen and Shuyu Xu are with University of Houston

# An Image-Space Morse Decomposition of Vector Fields\*

Guoning Chen<sup>†</sup> and Shuyu Xu<sup>†</sup>

## Abstract

Morse decompositions have been proposed to compute and represent the topological structure of steady vector fields. Compared to the conventional differential topology, Morse decomposition and the resulting Morse Connection Graph (MCG) is numerically stable. However, the granularity of the original Morse decomposition is constrained by the resolution of the underlying spatial discretization, which typically results in non-smooth representation. In this work, an Image-Space Morse decomposition (ISMD) framework is proposed to address this issue. Compared to the original method, ISMD first projects the original vector field onto an image plane, then computes the Morse decomposition based on the projected field with pixels as the smallest elements. Thus, pixel-level accuracy can be achieved. This ISMD framework has been applied to a number of synthetic and real-world steady vector fields to demonstrate its utility. The performance of the ISMD is carefully studied and reported. Finally, with ISMD an ensemble Morse decomposition can be studied and visualized, which is shown useful for visualizing the stability of the Morse sets with respect to the error introduced in the numerical computation and the perturbation to the input vector fields.

## Index Terms

Vector fields, Morse decomposition, image-space, stability analysis.

## I. INTRODUCTION

Vector field visualization and analysis has been applied to a wide variety of scientific and engineering applications to help domain experts interpret and present their data stemming from the study of various aero- and hydrodynamical systems. Example applications include aircraft and automobile engineering, astrophysics, plasma physic, mechanical engineering, oceanography, earthquake engineering, climate study and weather modeling, and medical practices. There are two types of visualization techniques for vector field data: 1) methods that generate densely and likely uniformly distributed visual primitives aligned with the vector field direction, e.g., glyphs, texture, and geometric-based visualization; 2) feature-based methods that first extract a small (and sparse) set of features or patterns of the flow and visualize them on top of the images generated in 1). The typical features that are extracted include topology, vortices, flow separation, shear flow, and other flow deformation described by the velocity gradient tensor, i.e., Jacobian. The feature-based methods are now getting more and more attention, as they automate the mental construction process that the domain experts have to perform to identify the features of interest from the images generated by non-feature-based methods. In addition, feature-based methods help achieve a reduced representation of the data to effectively present the results.

Among all the flow features that one cares about, vector field topology can be defined and computed without requiring a user-specified parameter. Vector field topology consists of various flow recurrences, including fixed points and periodic orbits and their connectivity. It partitions the flow domain into a number of regions. The flow within each region has homogeneous behaviors, i.e., all particle trajectories (i.e., streamlines) in the same region have the same starting point and the same ending point. In addition, all streamlines do not intersect with the boundary of the region. That said, this partition condenses the whole vector field into the boundaries of different regions, i.e., the topology, thus, an efficient representation and interpretation of the flow behavior can be obtained. Furthermore, it provides measurement of the sizes of the features and a rigorous neighboring relation between features, facilitating the subsequent consistent multiscale interpretation of the flow behavior. Therefore, vector field topology has become a popular method for analyzing steady vector fields.

\*This work was supported by the National Science Foundation, IIS-1352722.

<sup>†</sup>Guoning Chen and Shuyu Xu are with University of Houston

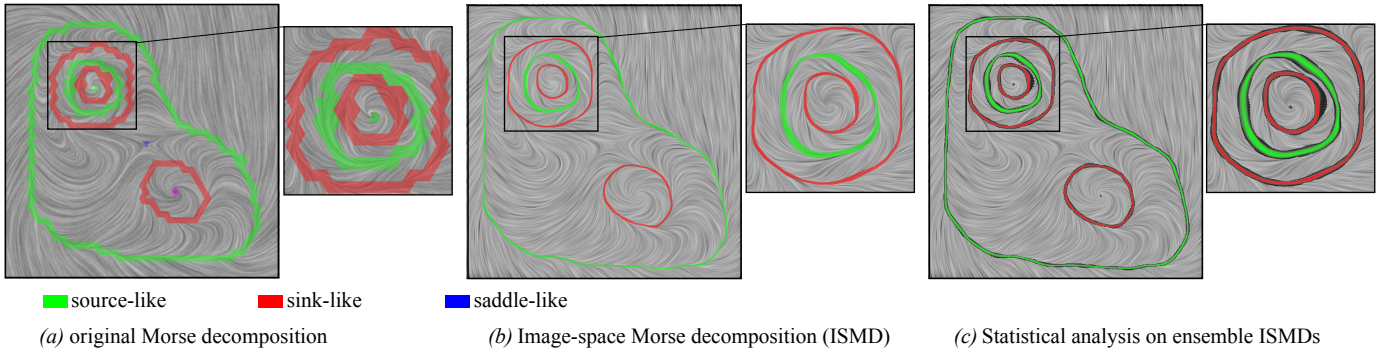


Fig. 1. The object-space Morse decomposition with  $\tau = 0.2$  (left), our image-space Morse decomposition (middle), and an ensemble of the image-space Morse decomposition for the stability study of the Morse sets (right), of a synthetic data [5].

Conventional methods resort to numerical integration to compute vector field topology, e.g., ECG and the topological skeleton, which can be easily compromised by the inherent numerical error, leading to unreliable interpretation of the flow structure. Recently, Morse decomposition has been introduced by Chen et al. [1] as a stable representation of the vector field topology. The idea of Morse decomposition is to compute a discrete topology that consists of Morse sets and their connectivity, i.e., the Morse Connection Graph (MCG). A Morse set is a polygonal element strip (e.g., a triangle strip for a triangulation of the domain) that *encloses* the flow recurrent dynamics as defined in the conventional topology. This coarse representation offers an additional room for each feature to tolerate certain amount of error or perturbation, resulting in a more stable representation compared to the conventional vector field topology. Considering its advantage, it has invoked a line of similar work [2], [3], [4] since its introduction.

However, there are a number of limitations of the original Morse decomposition. First, the resolution of the decomposition, i.e., the granularity or size of the Morse sets is constrained by the resolution of the underlying mesh. That is, the finest Morse set cannot be smaller than a triangle. This usually results in Morse sets with zig-zag like boundaries (Figure 1 (a)). This visually non-smooth representation is against the smooth nature of the flow. Second, the computation of Morse decomposition requires the adaptive re-sampling of particles along the boundaries of each polygonal element, leading to slow performance when processing large-scale data. This is because the adaptive framework is not easy to parallelize. Third, the existing computation and visualization of Morse decomposition does not provide the information of the stability of the Morse sets under different sources of uncertainty, such as the error introduced in the numerical integration and the perturbation and noise in the data. This stability information may help the domain experts to evaluate where are the most likely locations for the actual topological features (i.e., some special trajectories) within the detected Morse sets (i.e., some discrete regions). This important information is not well-conveyed by the original Morse decomposition and its visualization, because all points within a Morse set are shown with a uniform color (Figure 1 (a)), indicating the features can be anywhere in the Morse sets with equal chance, which is not true.

To address these limitations, we introduce an *Image-space Morse decomposition (ISMD)* framework. Similar to other image-space methods, we first *project* the original vector field that is defined in the object space onto an image plane. We then perform Morse decomposition based on the projected vector field. In contrast to the conventional Morse decomposition, ISMD estimates the flow maps by densely releasing particles within the individual pixels (i.e., some small squares) instead of resorting to the adaptive sampling along the edges. This new framework enables the parallel implementation of the flow map computation. The obtained directed graph represents the flow mapping between pixels, and thus, the extracted Morse sets are comprised of set of pixels. Consequently, ISMD typically achieves pixel-level accuracy compared to the original method (Figure 1 (b)). In addition, the performance of ISMD is independent of the size of the input data, thus, it is scalable to large-scale data sets. Finally, the ISMD framework enables us to develop an ensemble analysis framework for the study and visualization of the stability of the individual Morse sets with respect to different uncertainty sources (Figure 1 (c)). We carefully study and report the performance of the ISMD framework with respect to 1) the resolution of the image plane, 2) number of particles in each pixel, and 3) the integration steps. We also apply the proposed ISMD framework for the analysis of a number of synthetic and real-world vector field data to demonstrate its utility.

In the meantime, we wish to point out that ISMD shares the similar limitations as other image-space methods, e.g. view point dependency. In addition, it is equivalence to refining the underlying mesh to improve the estimation of the flow map, while replacing the more accurate but slower adaptive edge sampling strategy with a uniform dense sampling for particle placement to achieve faster computation. Hence, the accuracy of the obtained graph representation may not be guaranteed when compared to the original method. Therefore, instead of replacing the original Morse decomposition, we consider the ISMD a useful addition to the topology-based vector field analysis for efficiently generating an *overview* of the flow structure.

In the following, we will first review the related work in the literature of vector field visualization and analysis, and briefly describe the original Morse decomposition. Then, we will describe our ISMD framework and its visualization. Third, we apply it to a number of synthetic and real-world data sets and discuss the effectiveness of its utility. Fourth, we elaborate our framework of the ensemble analysis based on the ISMD and discuss its application for the uncertainty analysis and visualization for various flow data. Finally, we conclude our work with a discussion on its limitations and the future work.

## II. RELATED WORK

Vector field analysis is of primary interest in many areas of science and engineering. There are a vast variety of visualization and analysis techniques available these days for largely different vector-valued data. A series of survey papers [6], [7], [8], [9], [10], [11] provide a comprehensive review of this literature. In the following, we review only the most related work to the presented research.

Vector field topology was first introduced by Helman and Hesselink [12] to the visualization community. They defined a *topological skeleton* that consists of first-order fixed points and their connectivity. This topological skeleton has been extended to handle the boundary features [13] and higher-order fixed points [14], respectively. Wischgoll and Scheuermann [15] introduced a first solution to detecting periodic orbits from 2D steady vector fields, which has been extended to 3D vector fields [16]. Theisel et al. [17] proposed a grid-independent method for the location of periodic orbits. Chen et al. [5] introduced an efficient method for periodic orbit detection from 2D/2.5D vector fields and defined a more complete vector field topology by including periodic orbits into the topology construction. In the meantime, saddle-saddle connector has been introduced by Theisel et al. [18] for 3D vector field topology visualization to reduce the occlusion issue. Weinkauff et al. [19] proposed to extract high-order critical points to assist the simplification of 3D vector field topological representation.

Realizing the numerical instability issues of the previous methods that focus on the computation of the differential topology, Chen et al. [1] introduced the first stable and discrete representation of vector field topology based on Morse decomposition. This framework has been applied to 3D vector fields [20] and extended to construct a hierarchical representation of the flow structure [21], respectively. Recently, Szymczak and Zhang introduced a stable Morse decomposition framework for piecewise constant vector fields [4]. The presented work improves the conventional Morse decomposition for a smooth representation of vector field topology with the probability information encoded.

## III. VECTOR FIELD BACKGROUND AND MORSE DECOMPOSITION

In this section, we will briefly review some important concepts of vector fields and Morse decompositions.

Consider a  $d$ -manifold  $\mathbb{M} \subset \mathbb{R}^d$  ( $d = 2, 3$  in our cases), a general vector field can be expressed as an ordinary differential equation  $\dot{\mathbf{x}} = V(\mathbf{x}, t)$  or a map  $\varphi : \mathbb{R} \times \mathbb{M} \rightarrow \mathbb{R}^d$ , satisfying  $\varphi_{t_0}^{t_0}(\mathbf{x}) = \mathbf{x}_0$  and  $\varphi_0^{t+s}(\mathbf{x}) = \varphi_s^{t+s}(\varphi_0^s(\mathbf{x})) = \varphi_t^{t+s}(\varphi_0^t(\mathbf{x}))$ . This flow map describes the spatial correlation of points through *trajectories* (or flow paths) starting at time  $t_0$ :  $\varphi_{t_0}^t(\mathbf{x})$ . In steady flows, a point  $x_0 \in \mathbb{M}$  is a *fixed point* if  $\varphi(t_0, x) = x$  for all  $t \in \mathbb{R}$ . That is,  $V(x_0) = 0$ .  $x$  is a *periodic point* if there exists  $T > 0$  such that  $\varphi(T, x) = x$ . The trajectory of a periodic point is called a *periodic orbit*. Both fixed points and periodic orbits are examples of *flow recurrent features*. The connectivity of these flow recurrent features represents the qualitative (or structural) information of the vector fields, which we define as *vector field topology*.

*a) Morse Decomposition:* A Morse decomposition is a collection of disjoint closed sets (called Morse sets) that together contain all the recurrent dynamics of the flow induced by the vector field. More precisely, sets  $M_i$ ,  $i \in \{1, 2, \dots, N\}$  form a Morse decomposition if and only if the trajectory of any point is either (i) entirely contained in one of the Morse sets or (ii) is contained in some Morse set  $M_i$  for large enough negative times and in some



other Morse set  $M_j$ , with  $j > i$ , for large enough positive times. Intuitively, (ii) means that the trajectory of any point outside the Morse sets can only move from a set with lower subscript to a set with a higher subscript. (ii) excludes any recurrent dynamics outside the Morse sets, making it gradient-like [22]. In practice, the partial order between Morse sets can be represented as an acyclic directed graph called *Morse connection graph*, or MCG.

An indexing of Morse sets consistent with the above definition can be obtained from the MCG by means of topological sort. For an example shown in Figure 2, the Morse sets forming a Morse decomposition can be  $M_1 = R_1$ ,  $M_2 = S_1$ ,  $M_3 = S_2$ ,  $M_4 = S_3$ ,  $M_5 = A_2$ ,  $M_6 = A_1$ . Clearly, the MCG contains more restrictions on connecting trajectories than the resulting sequence (linear ordering) of Morse sets and therefore carries more information about the structure of the flow.

Chen et al. [1] describe a pipeline for the computation of Morse decompositions of the given vector fields (see the left flow chart in Figure 3). In this pipeline, the input vector field is firstly converted into a directed graph (i.e.  $FG$ ), denoted by  $\mathcal{F}$ , through a numerical computation, called flow combinatorialization. The nodes of  $\mathcal{F}$  are the individual triangles of the mesh where the vector field is defined. The directed edges indicate the flow mapping relations between triangles. For instance, if there is a directed edge  $T_1 \rightarrow T_2$ , the particles inside triangle  $T_1$  can be advected by the flow and enter  $T_2$ . In other words,  $\mathcal{F}$  encodes the dynamics of the flow at a combinatorial level. There are two approaches to compute  $\mathcal{F}$ : the geometry-based approach [5] and the  $\tau$ -map approach [1]. The geometry-based method computes the flow mapping (directed edges) between neighboring triangles by considering the flow behavior across each triangle edge. In contrast, the  $\tau$ -map approach keeps track of the image of each triangle over a constant time  $\tau$  to obtain the flow mapping.

Second, the strongly connected components are extracted from the directed graph  $\mathcal{F}$ . These strongly connected components correspond to the regions enclosing flow recurrence. The Conley index of each region is computed. Those regions with non-trivial Conley indices contain the Morse sets of interest [23]. The strongly connected components consisting of more than two triangles are also considered since they may contain multiple features that cancel each other such that the flow at the boundary acts like a regular flow.

Third, each strongly connected component corresponding to the region of non-trivial Conley index or region with more than one triangle is collapsed into a single node. This reduces the original graph into a quotient graph. From the quotient graph, the final MCG can be computed through path search between these strongly connected components using standard graph search algorithms.

The complete algorithm of this pipeline is provided in [1]. The limitation of this original method is the non-smooth boundaries of the obtained Morse sets in the visualization and the lack of the ability for the stability analysis.

#### IV. AN IMAGE-SPACE MORSE DECOMPOSITION

To address the limitations of the original Morse decomposition, we introduce an image-space Morse decomposition (ISMD). Different from the conventional Morse decomposition that is applied to the vector field defined in the physical (or object) space. The ISMD applies to the vector field defined in an image plane,  $I$ . Therefore, the graph nodes of the obtained directed graph,  $\mathcal{F}_I$ , from the flow combinatorialization correspond to the pixels (i.e., some small quadrilateral cells) of the image space instead of the polygonal elements in the object space.

##### A. Pipeline of the ISMD

b) *Projection to an image plane,  $I$* : To compute the vector field defined on an image plane,  $\mathcal{V}_I$ , from the original vector field, we resort to a vector field projection process that encodes the vector field into a colored image. In particular, the R, G, B color values of each pixel encode the x, y, z components of the vector value defined at the center of that pixel. For a 2D planar vector field, only the first two color channels are utilized. The following

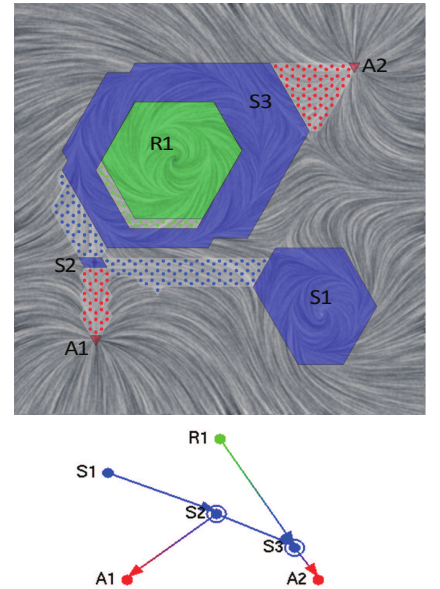


Fig. 2. An MCG of a synthetic vector field [21].

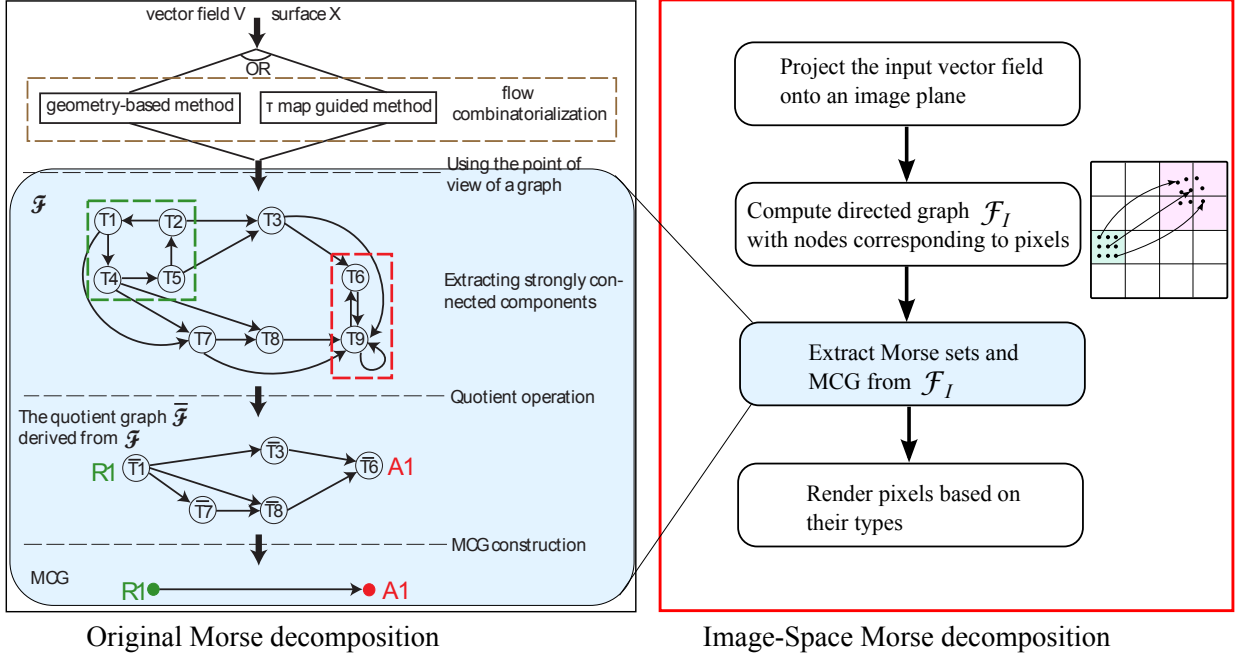


Fig. 3. The original Morse decomposition computation pipeline (left) versus the proposed image-space Morse decomposition (right).

formula describe the encoding of the vector value into R, G, B colors.

$$R = \frac{vx - vx_{min}}{vx_{max} - vx_{min}}; \quad G = \frac{vy - vy_{min}}{vy_{max} - vy_{min}}; \quad B = \frac{vz - vz_{min}}{vz_{max} - vz_{min}};$$

where  $vx_{min}$  is the minimum value of the x component of the vectors over the domain, and  $vx_{max}$  is the maximum. The decoding of the vector value from the R, G, B colors can be computed as follows.

$$vx = vx_{min} + R(vx_{max} - vx_{min}); \quad vy = vy_{min} + G(vy_{max} - vy_{min}); \quad vz = vz_{min} + B(vz_{max} - vz_{min});$$

c) *Flow combinatorialization in the image plane:* After getting  $\mathcal{V}_I$ , the flow combinatorialization can be computed by tracking the image of each pixel,  $\mathbf{p}_i$ , over a given time  $\tau$ . To do so, we uniformly sample each pixel with  $n_{sd}$  particles and trace them by following the vector field  $\mathcal{V}_I$  over certain steps,  $n_{stp}$ , and see where they end. A directed edge is then added to the directed graph  $\mathcal{F}_I$  pointing from the starting pixel  $\mathbf{p}_i$  to each pixel  $\mathbf{p}_j$  that contains the image of a particle released from  $\mathbf{p}_i$ . Without specifically mentioning, the results shown in this paper are computed using a Runge-Kutta second-order (RK2) integrator with a constant step size. As  $\mathcal{V}_I$  is defined on an image plane, it can be represented as a texture. Hardware acceleration techniques, such as OpenMP and CUDA can be utilized to efficiently integrate the trajectories of the released particles. In this work, we focus only on the CPU implementation of the ISMD.

Note that the above computation simplifies the adaptive edge sampling algorithm for the estimation of the image of the polygonal elements in the original Morse decomposition. However, in order to capture the divergence and stretching behavior of the flow and maintain a simply-connected image of a pixel, a large number of particles should be released. In a later discussion, we will study how different numbers of sampled particles within a pixel affect the ISMD results.

d) *Extracting Morse sets and constructing MCG:* After getting the directed graph  $\mathcal{F}_I$ , the Morse sets can be similarly extracted as the strongly connected components of  $\mathcal{F}_I$ . And the subsequent MCG can be constructed by searching the paths between neighboring Morse sets. Both steps remain the same as the original Morse decomposition computation. We follow the same classification criteria as the original method to classify the detected Morse sets. Specifically, a Morse set is a *source-like* Morse set (colored in green) if its corresponding sub-graph does not have any edges pointing from the nodes outside of the sub-graph to the nodes in the interior. A Morse set is *sink-like* (colored in red) if its corresponding sub-graph does not have any edges pointing from the nodes of the sub-graph to the nodes outside of it. A Morse set is *saddle-like* (colored in blue) if it is neither a source-like

Morse set, nor a sink-like Morse set.

e) *Effect of different numbers of samples*: According to the description above, the user needs to determine a proper number of samples within each pixel in order to most accurately capture the image of the pixel and construct an accurate graph,  $\mathcal{F}_I$ . However, determining the right number of samples and how they are distributed is not trivial. This is because we are seeking for two conflict goals. On the one hand, we want to place as many samples as possible within a pixel for accuracy concern. On the other hand, we want a fast computation. In the current work, we resort to a simple uniform sampling strategy and place the same amount of samples within all pixels. Figure 4 shows a study of how different numbers of samples affect the obtained ISMDs. We use an oceanic current around the Iceland for this study. This data set consists of 23,655 triangles. We show the ISMD results with 4, 9, and 36 samples, respectively. The result with only 4 samples (left-most image of Figure 4) contains a lot of Morse sets with non-smooth boundaries and missing interior pixels (i.e., not simply-connected). With the increase of the number of samples, we obtain more and more well-connected Morse sets with smoother boundaries. However, after releasing more than 9 samples within a pixel, the ISMD result does not improve a lot. This may indicate that 9 samples are sufficient for this data set. As expected, with the increase of the number of samples, the computation time increases, as shown in the upper-right plot of Figure 4. Further discussion on the performance of ISMDs with respect to the number of samples will be provided in Section IV-B.

To explain why a uniform sampling strategy is adopted in our work, we show an ISMD result with 30 random samples within each pixel in the lower-bottom image (highlighted by the red box) of Figure 4. As we can see, many of the obtained Morse sets have disconnected regions and non-smooth boundaries. This is because the locally random sampling strategy cannot capture the spatially coherent flow behavior. As discussed in the original Morse decomposition computation [1], the number of samples that is needed is determined by the divergence of the flow during the particle advection, i.e., more samples will be inserted if the distance between pairs of neighboring samples increases during the advection. This flow divergence behavior can be measured by the rate of flow separation – a quantity computed using the *Finite Time Lyapunov Exponent (FTLE)* [24]. Therefore, a more efficient and adaptive sampling strategy, such as those for the FTLE computation [25], can be adopted to improve our current sampling. We plan to investigate this in the future.

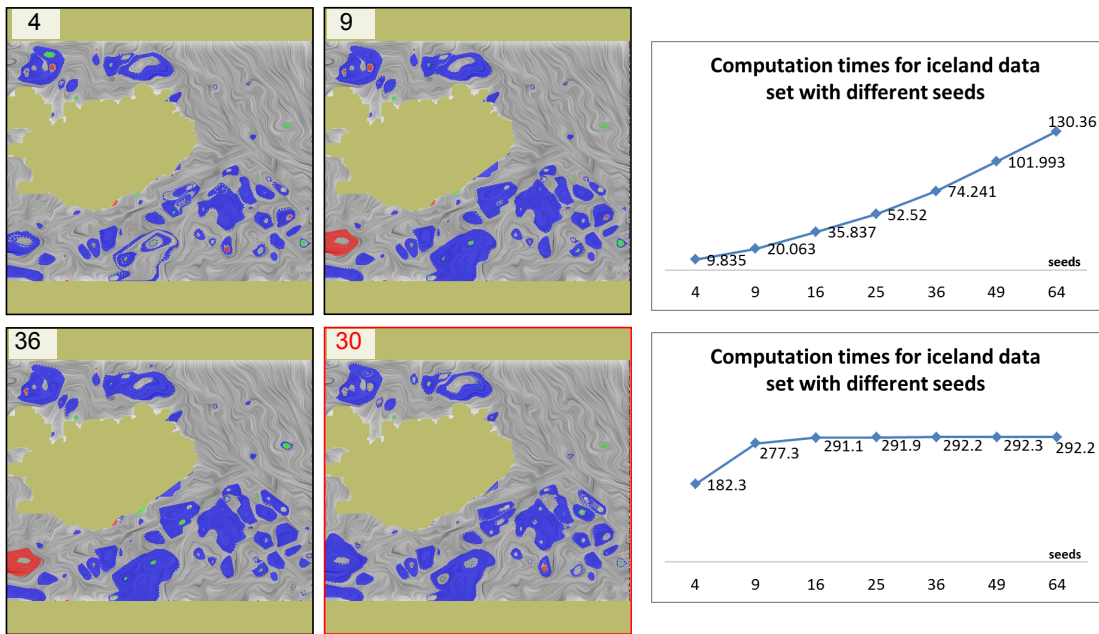


Fig. 4. ISMD results of the Iceland data with different uniformly distributed samples within a pixel from 4 to 49. These results are computed with 512 resolution and 5 integration steps. The plot shows the times spent and memory consumptions of the regular sample results. The image highlighted with a red rectangle shows the result with 30 random seeds within a pixel.

TABLE I

PERFORMANCE OF THE ISMDs OF THE ICELAND CURRENT DATA WITH DIFFERENT NUMBERS OF SAMPLES WITHIN EACH PIXEL  
( $res = 512^2$ ,  $n_{stp} = 5$ ).

| Measurement \ $n_{sd}$                           | 4       | 9       | 16        | 25        | 36        | 49        | 64        |
|--|---------|---------|-----------|-----------|-----------|-----------|-----------|
| Time (s)   | 9.835   | 20.063  | 35.837    | 52.52     | 74.241    | 101.993   | 130.36    |
| Memory (MB)                                      | 182.3   | 277.3   | 291.1     | 291.9     | 292.2     | 292.3     | 292.2     |
| #Morse_sets                                      | 132     | 128     | 126       | 121       | 121       | 116       | 116       |
| #edges in $\mathcal{F}_I$                        | 821,877 | 942,675 | 1,002,455 | 1,033,304 | 1,054,481 | 1,072,156 | 1,085,724 |
| $k = \frac{\#edges}{\#nodes}$ in $\mathcal{F}_I$ | 3.135   | 3.596   | 3.824     | 3.942     | 4.022     | 4.09      | 4.142     |

TABLE II

PERFORMANCE OF THE ISMDs OF THE ICELAND CURRENT DATA WITH DIFFERENT IMAGE RESOLUTIONS ( $n_{stp} = 5$ ,  $n_{sd} = 16$ ).

| Measurement \ res                                | 64     | 128    | 256     | 512       | 1024      |
|--|--------|--------|---------|-----------|-----------|
| Time (s)   | 0.644  | 2.277  | 9.071   | 35.837    | 140.033   |
| Memory (MB)                                      | 45.6   | 58.1   | 104.1   | 291.1     | 1038.6    |
| #Morse_sets                                      | 34     | 49     | 88      | 126       | 147       |
| #edges in $\mathcal{F}_I$                        | 20,785 | 74,083 | 269,741 | 1,002,455 | 3,815,585 |
| $k = \frac{\#edges}{\#nodes}$ in $\mathcal{F}_I$ | 5.074  | 4.522  | 4.116   | 3.824     | 3.639     |

TABLE III

PERFORMANCE OF THE ISMDs OF THE ICELAND CURRENT DATA WITH DIFFERENT INTEGRATION STEPS ( $res = 512^2$ ,  $n_{sd} = 16$ ).

| Measurement \ $n_{stp}$                          | 2       | 5         | 8       | 10        | 15      | 20        | 25        |
|--|---------|-----------|---------|-----------|---------|-----------|-----------|
| Time (s)   | 20.888  | 35.837    | 51.383  | 61.975    | 87.167  | 113.137   | 138.478   |
| Memory (MB)                                      | 287.2   | 291.1     | 291.4   | 293.4     | 294.1   | 296.4     | 297.2     |
| #Morse_sets                                      | 56      | 126       | 143     | 159       | 155     | 148       | 142       |
| #edges in $\mathcal{F}_I$                        | 902,604 | 1,002,455 | 858,018 | 1,082,193 | 975,811 | 1,198,893 | 1,247,442 |
| $k = \frac{\#edges}{\#nodes}$ in $\mathcal{F}_I$ | 3.443   | 3.824     | 3.273   | 4.128     | 3.722   | 4.573     | 4.759     |

## B. Results

Figure 7 shows the image-space Morse decomposition results of a number of synthetic and real-world 2D vector field data sets. For comparison, their corresponding object-space Morse decompositions are also provided. In general, we see that the ISMD results extract Morse sets with much smoother boundaries, compared to the object-space method. A detailed comparison will be provided later.

*f) Performance:* To study the performance of the proposed image-space Morse decomposition, we conduct a number of experiments using the oceanic currents around Iceland. There are a number of factors that may affect the performance (e.g., computational time and memory consumption) of the ISMD. In these experiments, we particularly concentrate on three factors: 1) size of the image plane,  $res$  2) number of integration steps,  $n_{stp}$  and 3) number of samples within one pixel,  $n_{sd}$ . Tables I, II, III provide the performance report of this study, respectively. All the performance information is measured on a PC with Intel Xeron 1.6GHz quad-core processor and 8GB RAM without parallel computation. Note that the object-space Morse decomposition takes about 263 seconds to extract 86 Morse sets with  $\tau = 0.15$  for this data set, and its memory footprint is 108 MB. The obtained directed graph has 257,415 edges.

To see how different sizes of the image plane affect the performance, we compute the ISMDs for the Iceland current data with resolutions 64, 128, 256, 512, and 1024, respectively. Note that in all our experiments, we assume a square-like image plane. Therefore, without further specification the number of pixels of the image with resolution

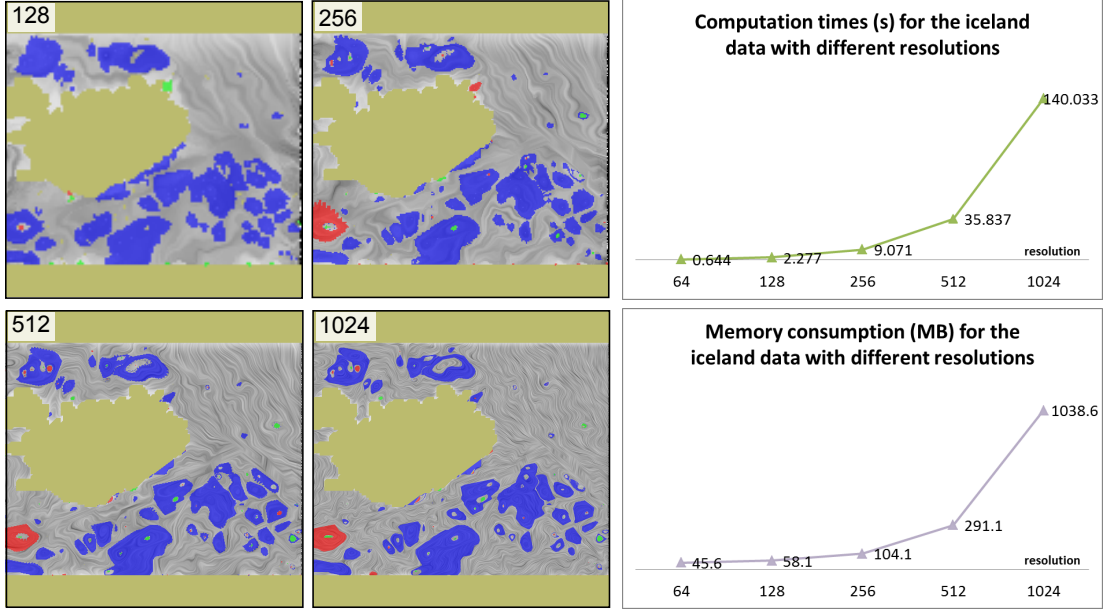


Fig. 5. ISMD results of the Iceland data with different resolutions from 128 to 1024. These results are computed with a regular sampling with 16 seeds per pixel and 5 integration steps. The right most column shows the times spent and memory consumptions of these results.

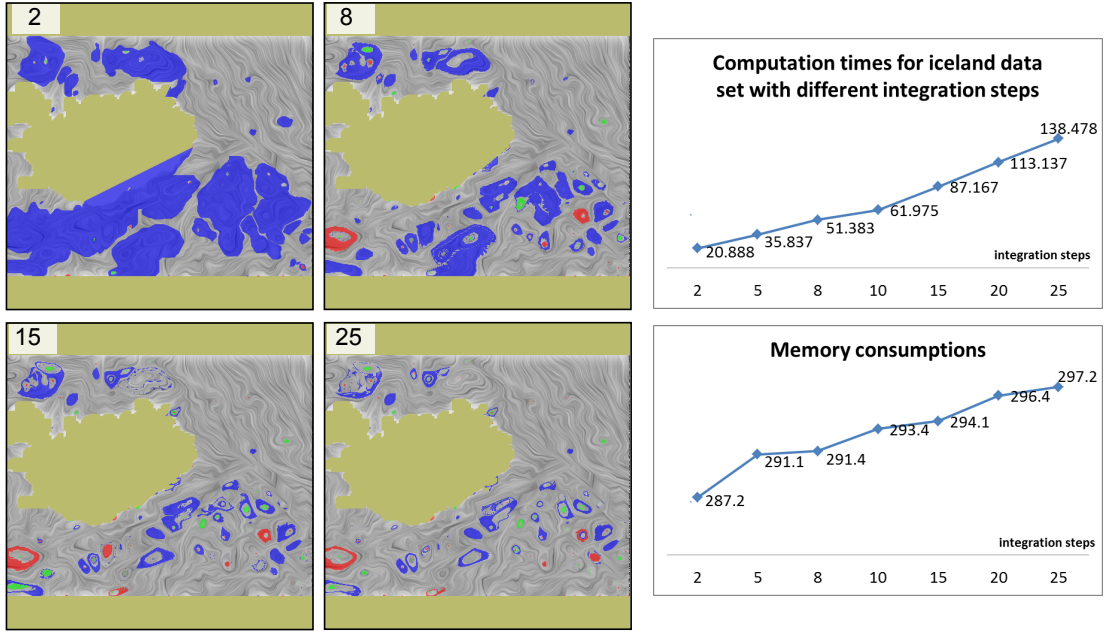


Fig. 6. ISMD results of the Iceland data with different integration steps from 2 to 25. These results are computed with 512 resolution and a regular sampling with 16 seeds per pixel. The right most column shows the times spent and memory consumptions of these results.

$res$  is  $res^2$ . In all these ISMD computations, we fix  $n_{stp} = 5$  and  $n_{sd} = 16$ . From the experiments, we observe that with the increase of the image size, both the computation and memory costs increase accordingly, especially, the complexity of computation cost is linear with respect to the image resolution. This is expected as the computation time is determined by the number of particles that are released in the image plane, and the memory consumption is determined by the size of the graph,  $\mathcal{F}_I$ . Both the number of the particles and the size of  $\mathcal{F}_I$  are functions of the image resolution,  $res$ . Specifically, the number of the particles that are released is  $res \times n_{sd}$  and the graph size  $|\mathcal{F}_I| \approx res^2 + k \times res^2$ , where  $k = \frac{\#edges}{\#nodes} \in \mathbb{R}^+$  is a positive constant that measures the average number of edges adjacent to a node in  $\mathcal{F}_I$ . Figure 5 shows the ISMDs with different resolutions and the corresponding plots of their performance. The last row of Table II shows the average number of edges, i.e.,  $k$ , adjacent to each node in



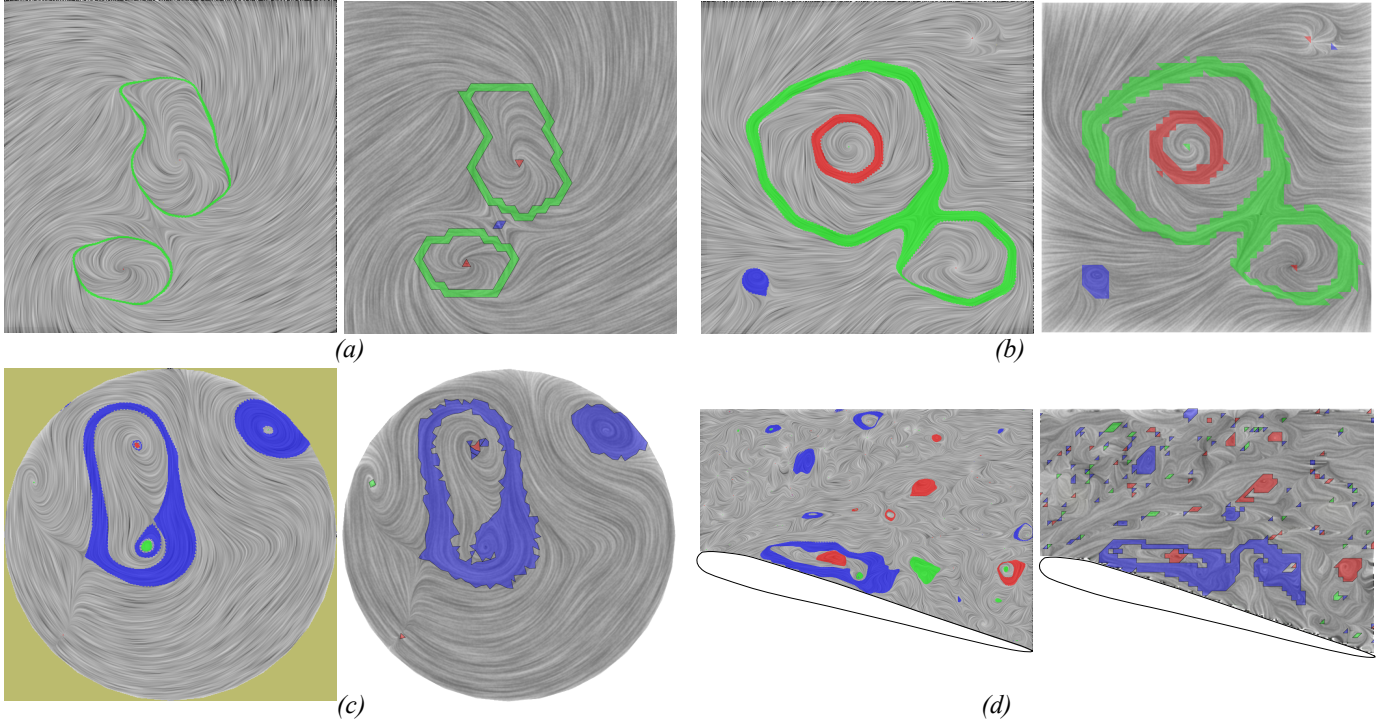


Fig. 7. ISMDs (left) of a number of vector fields and their object-space Morse decompositions (right): (a)-(b) two synthetic vector fields; (c) a slice of the diesel engine simulation [5]; (d) a time step of the OSU wind tunnel data [26].

the obtained directed graph  $\mathcal{F}_1$ . As we see that  $k$  decreases when the resolution increases, which indicates that the graph has been refined, i.e., one node is split to multiple ones, accordingly.

Figure 6 shows the ISMDs with different integration steps, i.e., different integration times. In this study, we fix  $res = 512$  and  $n_{sd} = 16$ . From the obtained results, we see that with the increase of the integration steps (or times), the obtained Morse sets become finer. This important property has been shown in the Morse decomposition computed in the object-space [1], and has been applied to achieve multiscale representation of the flow structure [21]. Similarly, with the increase of the integration steps, the computation time increases linearly, while the increase of memory consumption is much slower.

The effect of the number of samples within each pixel and the sampling strategy has been discussed previously. How different numbers of samples affect the performance is illustrated in the plots of Figure 4. Again, with the increase of the number of samples within each pixel, the computational time increases linearly based on the discussion above, while the memory consumption does not change much after increasing the number of samples above 9, meaning that the graph does not improve a lot with more samples. This indicates that for this data set 9 samples per pixel should be sufficient. A similar observation can be seen for other incompressible or weak-incompressible flow data, for instance, the combustion data as shown in Figure 8. This is a 2D time-varying vector field simulation of a homogeneous charge compression ignition (HCCI) engine combustion [27] represented as a  $640 \times 640$  regular grid with a periodic boundary. The data set consists of 299 time steps. We use step 137 in our experiment. The ISMD results do not improve much with the increase of the samples after 9 or more samples are used. Note that Table I also shows that the average number of the edges adjacent to each node,  $k$ , increases with the increase of samples, indicating that the accuracy of the graph has been improved.

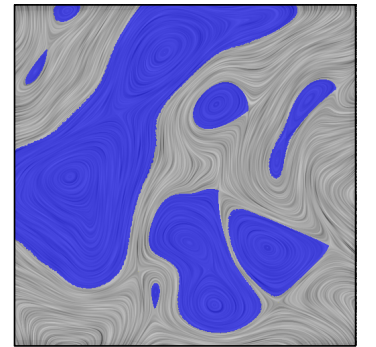


Fig. 8. ISMD of the HCCI data set ( $res = 512$ ,  $n_{stp} = 50$ ,  $n_{sd} = 16$ ).

### C. Comparison with the conventional Morse decomposition

As described before, image-space approach is suitable for providing a quick overview of the data set without worrying about the size of the original data. Therefore, its performance is



independent of the data size. This facilitates the analysis and visualization of some large scale data that the traditional method has trouble to handle. For instance, for the aforementioned HCCI data (Figure 8), the traditional object-space Morse decomposition cannot compute its MCG due to insufficient memory space, while the ISMD successfully computes its decomposition. The ISMD also provides smoother visualization of the detected Morse sets, which more closely reflect the smooth nature of the underlying flow behavior. In some cases, the ISMD locates more accurate position of the Morse sets, especially when they are close to the boundary of the domain where the sampling is not sufficient for an accurate computation in the object-space method. Figure 9 provides such an example. It shows a 2D slice of the ABC flow. The magnified view of the left image highlights a Morse set (i.e., a green triangle) that does not match the vortex center as indicated by the LIC texture. In contrast, our ISMD accurately locate and highlight this vortex area (i.e., the blue disk). As this Morse set contains a saddle, thus, its type is saddle-like and shown as blue.

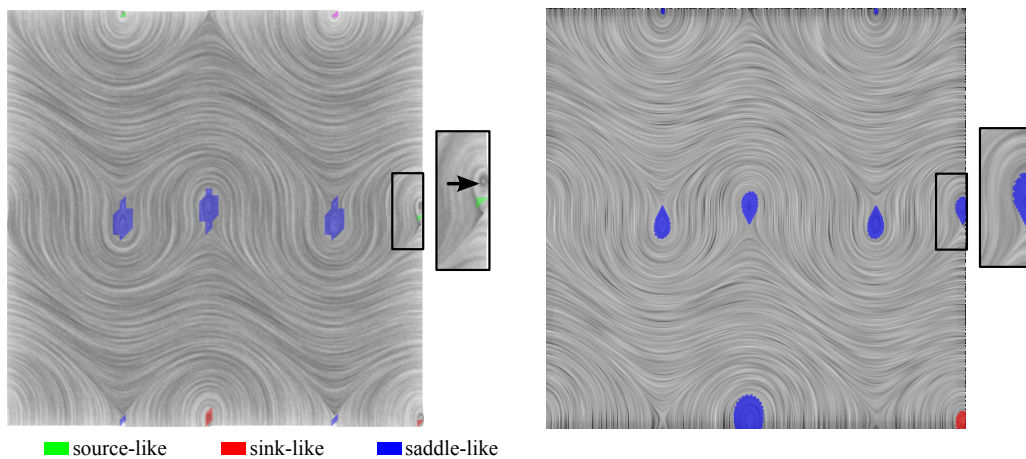


Fig. 9. The object-space Morse decomposition (left) and our ISMD (right) of a 2D slice of the ABC flow. The magnified view highlights a region in the flow where the object-space method fails to accurately locate the true Morse set (shown by the underlying LIC) at the boundary of the domain, while our ISMD can accurately identify and visualize.

We have also applied the ISMD to the analysis of two time-dependent vector fields. These two data sets are taken from the top layer of a 3D simulation of global oceanic eddies for 350 days of the year 2002 [28]. The 2D time-varying vector field has resolution  $3600 \times 2400$ . We extract tiles from the central Atlantic Ocean ( $60 \times 60$ ), denoted by *OceanA*, and south Atlantic Ocean ( $100 \times 100$ ), denoted by *OceanB*, respectively. We use slices from #20311 to #20314 from *OceanA*, and slices from #20606 to #20609 from *OceanB* for this experiment [28]. We apply ISMD to the individual time steps of the vector fields which are essentially some steady vector fields. The resolution of the ISMD for these two examples is 512. Figures 12 and 11 show the ISMD results (top row) of these two data sets for four consecutive time steps. For comparison, we show the traditional Morse decomposition results in the bottom row. The computation of the ISMD results takes less than 35 second per-frame, while the computation of the object-space Morse decomposition takes about 167 for *OceanA* and 268 seconds for *OceanB* per-frame. Overall, the transition of Morse sets from ISMD over time is much smoother than those from the traditional method.

*g) Difference between ISMD and object-space Morse decomposition:* To compare with the object-space Morse decomposition, we check whether a pixel that belongs to a Morse set from the ISMD falls in a triangle that is in a Morse set in the object-space, and vice versa. We record the difference in two types, i.e., I) pixels in an ISMD Morse set but not in an object-space Morse set, and II) pixels not in an ISMD Morse set but in an object-space Morse set. The former we consider it a false positive, while the latter a false negative. The pixel-triangle mapping relation can be represented by computing an additional image in the projection step of the ISMD pipeline. This image assigns a unique color to a pixel based on the triangle ID that this pixel falls in. Similar method has been used in the coherent texture-based flow visualization [29].

Based on the above difference types, we define a measurement to quantify the difference between the two approaches. Let  $n_I$  and  $n_{II}$  be the numbers of type I and type II pixels, respectively. We can define a number of error rate based on  $n_I$  and  $n_{II}$ : 1) type I error rate,  $err_I = \frac{n_I}{res^2}$ , 2) type II error rate,  $err_{II} = \frac{n_{II}}{res^2}$ , and 3) total error rate,  $err_{total} = err_I + err_{II}$ . Since we wish to capture as many recurrent features as possible during analysis, type

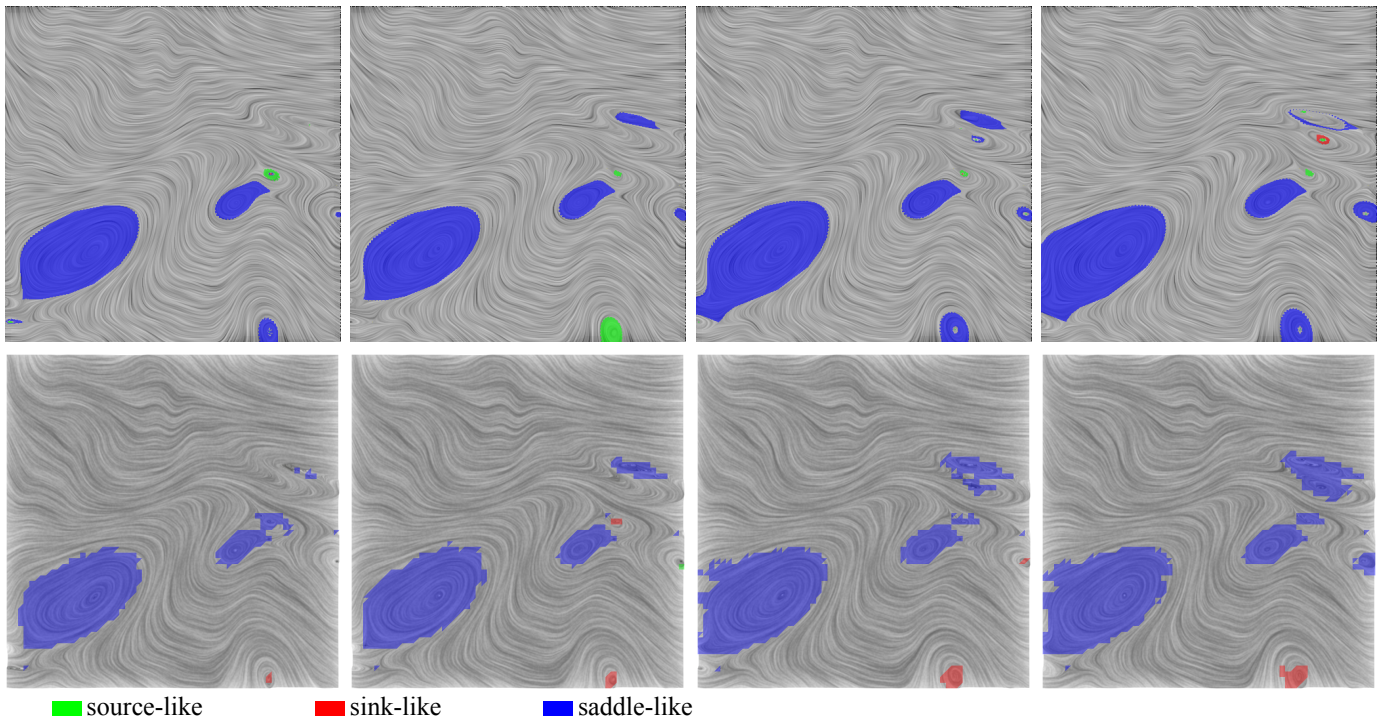


Fig. 10. *OceanA* data set from slice #20311 to #20314. Top row shows our ISMD results, while the bottom row shows the object-space Morse decomposition with  $\tau = 0.2$ .

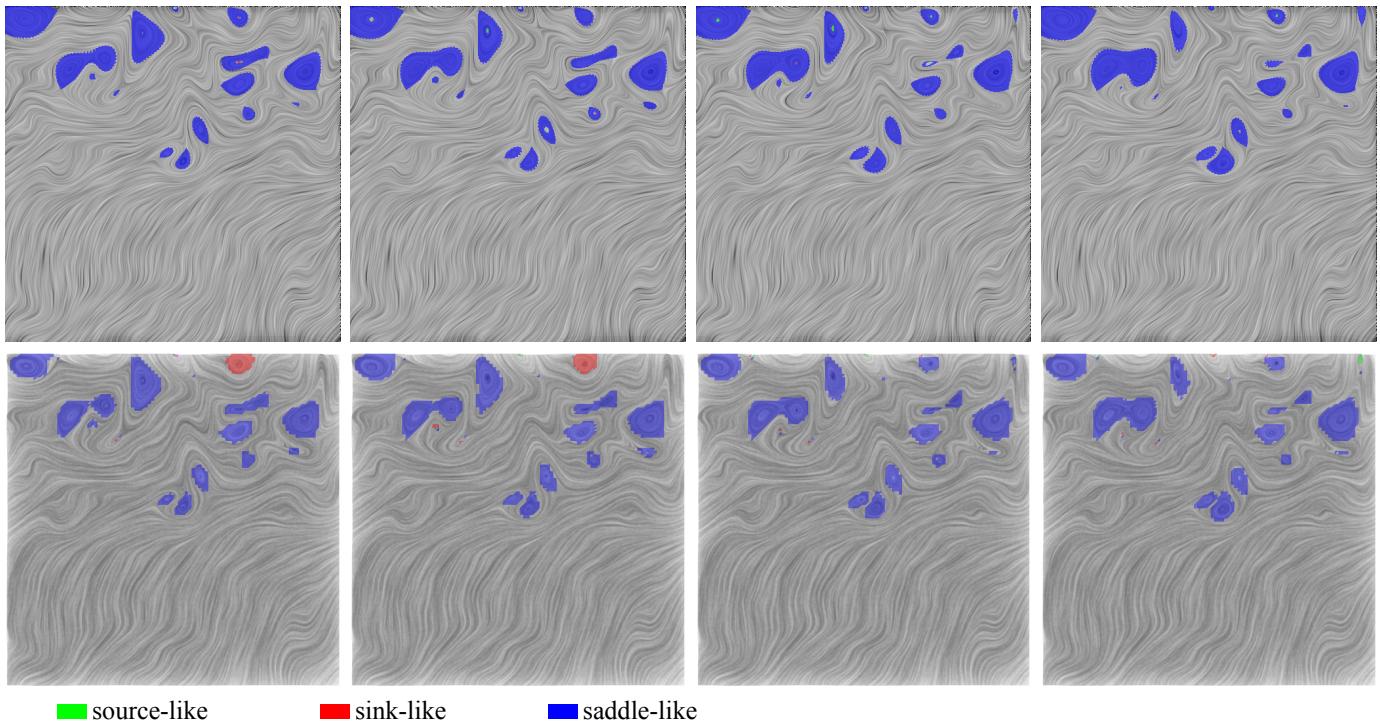


Fig. 11. *OceanB* data set from slice #20606 to #20609. Top row shows our ISMD results, while the bottom row shows the object-space Morse decomposition with  $\tau = 0.2$ .

II error (i.e., missing features) is more important when compared to the object-space results which are considered more accurate.

Table IV provides the statistics of the comparison of the ISMD and the traditional Morse decomposition via a

number of data sets. Note that all the ISMD results use  $res = 512$ ,  $n_{stp} = 5$ , and  $n_{sd} = 16$ . Table V provides the error measurement of the ISMDs of the Iceland current data with different numbers of samples when compared to an object-space Morse decomposition with  $\tau = 0.15$ . While the type I error increases with the increase of the numbers of samples within a pixel, the type II error decreases accordingly. This indicates that with more samples, the ISMD results capture more and more flow regions that have recurrent flows which are also identified by the object-space method. Visually, the Morse sets in the image space have less missing interior pixels and smoother boundaries with more samples. Table VI provides the error measurement of the ISMDs of the Iceland current data with different resolutions when compared to an object-space Morse decomposition with  $\tau = 0.15$ . The type I error decreases with the increase of the image resolution, while the type II error remains in a similar level with small fluctuation. The overall error rate generally decreases with the increase of the image resolution, indicating that the accuracy is improved with higher image resolution (i.e., finer sampling), as expected.

TABLE IV

ERROR STUDY OF THE ISMDs OF A NUMBER OF DATA SETS USED IN THIS WORK. ( $res = 512$ ,  $n_{stp} = 5$ ,  $n_{sd} = 16$ ). THE OBJECT-SPACE MORSE DECOMPOSITIONS OF THESE DATA SETS ARE TYPICALLY COMPUTED WITH  $\tau = 0.2$  EXCEPT FOR THE WIND TUNNEL DATA SET WHERE  $\tau = 0.5$  IS USED.

| Error \ Datasets | OceanA |      |      |      | OceanB |      |      |      | ABC  | Fig.1 | Fig.13 |
|------------------|--------|------|------|------|--------|------|------|------|------|-------|--------|
|                  | 311    | 312  | 313  | 314  | 606    | 607  | 608  | 609  |      |       |        |
| $err_I$          | 0.6%   | 0.5% | 1.7% | 1.1% | 2.0%   | 2.0% | 2.2% | 2.1% | 0.7% | 0.1%  | 1.5%   |
| $err_{II}$       | 1.2%   | 1.5% | 0.7% | 3.4% | 1.6%   | 1.9% | 1.7% | 1.7% | 0.4% | 0.4%  | 4.7%   |
| $err_{total}$    | 1.8%   | 2.0% | 2.4% | 4.5% | 3.6%   | 3.9% | 3.9% | 3.8% | 1.1% | 0.5%  | 6.2%   |

TABLE V

ERROR STUDY OF THE ISMDs OF THE ICELAND CURRENT DATA WITH DIFFERENT NUMBERS OF SAMPLES WITHIN EACH PIXEL ( $res = 512$ ,  $n_{stp} = 5$ ), COMPARED TO THE OBJECT-SPACE MORSE DECOMPOSITION WITH  $\tau = 0.15$ .

| Error \ $n_{sd}$ | 4    | 9    | 16   | 25   | 36    | 49   | 64   |
|------------------|------|------|------|------|-------|------|------|
| $err_I$          | 2.2% | 4.2% | 4.4% | 4.6% | 4.7%  | 5.1% | 5.6% |
| $err_{II}$       | 5.2% | 3.9% | 3.5% | 3.3% | 3.26% | 3.2% | 2.7% |
| $err_{total}$    | 7.4% | 8.1% | 7.9% | 7.9% | 7.96% | 8.3% | 8.3% |

TABLE VI

ERROR STUDY OF THE ISMDs OF THE ICELAND CURRENT DATA WITH DIFFERENT RESOLUTIONS ( $n_{stp} = 5$ ,  $n_{sd} = 16$ ), COMPARED TO THE OBJECT-SPACE MORSE DECOMPOSITION WITH  $\tau = 0.15$ .

| Error \ $n_{sd}$ | 64    | 128  | 256  | 512  | 1024 |
|------------------|-------|------|------|------|------|
| $err_I$          | 9.3%  | 6.3% | 5.1% | 4.4% | 4.1% |
| $err_{II}$       | 3.2%  | 2.7% | 3.4% | 3.5% | 4.0% |
| $err_{total}$    | 12.5% | 9.0% | 8.5% | 7.9% | 8.1% |

**Remark** It is worth noting that our ISMD can be combined with the traditional Morse decomposition framework in order to improve its ability of encoding important flow information. Since the pixel-triangle mapping relation is stored in the previous error estimation, it is straight forward to compute the directed graph in the object-space,  $\mathcal{F}_O$ , based on the results of the graph in the image space,  $\mathcal{F}_I$ . Particularly, if there is a directed edge  $p_i \rightarrow p_j$  in  $\mathcal{F}_I$ , and  $p_i \in T_k, p_j \in T_l$  ( $T_k$  and  $T_l$  are triangles of the original mesh), we add an edge  $T_k \rightarrow T_l$  in  $\mathcal{F}_O$ . In addition, we can store the statistic information in the flow map given the current triangulation as the edge weights. Specifically, given all pixels that fall in a triangle  $T_k$ , denoted by  $\{p\}|_{T_k}$ , we define the weight on the edge  $T_k \rightarrow T_l$  as  $\frac{|\{p\}|_{T_k \rightarrow T_l}|}{|\{p\}|_{T_k}}$ , where  $\{p\}|_{T_k \rightarrow T_l}$  is the set of pixels that can reach triangle  $T_l$  from  $T_k$ , which is non-empty, and  $|\cdot|$  represents the number



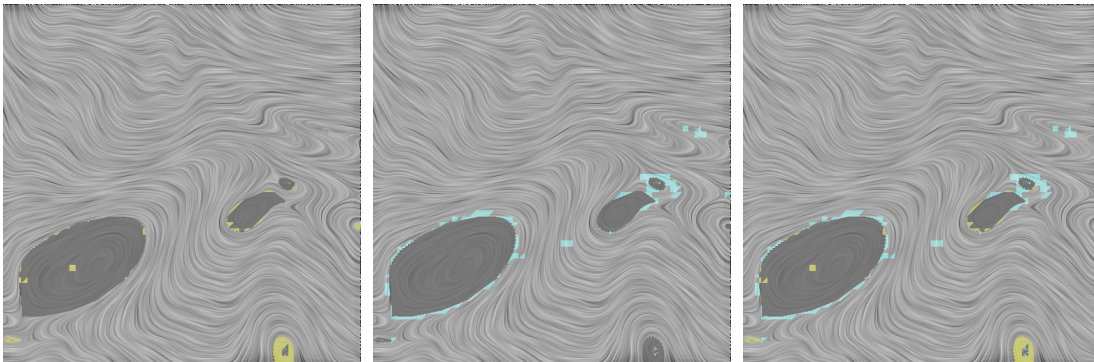


Fig. 12. The difference between the ISMD versus the object-space Morse decomposition on the slice #20311 from the *OceanA* data. There are 1,471  $n_I$  type of pixels (yellow in the left image) and 3,256  $n_{II}$  type of pixels (cyan in the middle image). The right image shows all the pixels that the two methods do not agree.

of pixels in a given set. With this statistical information encoded, a few graph analyses can be conducted [30], [31], which we plan to further explore in the future.

#### D. Discussion and Limitations

The aforementioned ISMD framework can be extended to 3D volumetric vector field analysis by resorting to the 3D texture representation. However, it currently does not apply to vector fields defined on curved surfaces. This is because even though any local neighborhood of a surface point behaves like a plane, without a proper surface parameterization, it is impossible to project a general surface vector field on to a planar image. To address that, one can consider the surface flow is constrained in a 3D thin shell that encloses the surface boundary. This approximates the surface vector field as a volumetric field that has non-zero value only within the thin shell. Similar idea has been applied by Kwatra et al. for texturing dynamic fluid surface [32]. Another alternative is to adapt the method on coherent texture-based flow visualization [29]. We plan to investigate this in a future work.

Another limitation of the ISMD is its view point dependency, which requires the recomputation of the ISMD whenever the user changes the view point, such as zooming and panning. In addition, if a portion of a circulating feature is outside of the image plane due to the user interaction, the ISMD will not be able to capture it, as the projected vector field will not be able to represent the whole feature. This is an inherent limitation of all the image-space methods. Therefore, instead of replacing the more accurate object-space computation, ISMD should serve as a tool to achieve quick overview of the flow recurrent dynamics before the accurate extraction.

### V. A STUDY OF MORSE SET STABILITY USING ISMD

With the proposed ISMD framework, the stability of the detected Morse sets under different sources of uncertainty can be approximated and visualized, which is difficult to obtain via the conventional Morse decomposition framework. In this study, we consider two sources of uncertainty, including 1) the error introduced during integration and 2) the noise contained in the vector field data. For both studies, we artificially introduce error and noise to the data and the computation, and examine the behavior of the obtained ISMD in an ensemble fashion.

#### A. Integration Error

In this study, we perform ISMD on the same vector fields with different numerical integrators and different integration step sizes, i.e., with different integration error. In particular, we experiment with the Euler, RK2, and RK4 integrators, respectively. Two random integer numbers,  $n1_{ran}$  and  $n2_{ran}$ , are computed before each ISMD computation, which are used to determine the integrator and the corresponding step sizes for this ISMD computation. This results in an ensemble of the ISMDs of the same vector field. For each of these ISMD results, we label each pixel as either 0 (not within a Morse set) or  $k \in \{1, 2, 3\}$  (1—in a source-like Morse set, 2—in a sink-like Morse set, or 3—in a saddle-like Morse set). Next, we count how many times a pixel falls in a Morse set,  $n_{hits}$ , and what is the type of the Morse set where it falls in. This information will be used to determine the color of the pixel in the

rendering. Specifically, we adopt the HSV color space. For each pixel, we assign the hue value as either 120 if most times the pixel is in a source-like Morse set; 0 if it is mostly in a sink-like Morse set; 240 if it is mostly in a saddle-like Morse set; or 330 if its type cannot be determined. The saturation and intensity values of the pixel are both set as  $\frac{n_{hits}}{N_{trials}}$ , where  $N_{trials}$  is the number of trials for this ensemble analysis. With this simple transfer function, a pixel that is more likely in a Morse set (i.e., larger  $n_{hits}$  value) will have brighter and more saturated color, while a pixel that has a low likelihood to be in a Morse set will get a darker color.

Figure 13 shows the ensemble analysis of the ISMD and the stability of the Morse sets under different integration schemes for two synthetic (left two images) and two real-world data sets (right two images). For these experiments,  $res = 512$ ,  $n_{stp} = 5$ ,  $n_{sd} = 16$ ,  $N_{trials} = 20$ ,  $n1_{ran} \in \{0, 1, 2\}$ , and  $n2_{ran} \in \{1, 2, \dots, 10\}$ . For example, if  $n1_{ran} = 0$ , the Euler integrator with step size  $dt = \frac{\Delta}{n2_{ran}}$  is selected, where  $\Delta$  is the interval between two neighboring pixels along X direction. From these results, we see for all these vector fields, the less stable areas of the Morse sets under different error introduced by the integration are near their boundaries (highlighted by the magenta arrows), while for the incompressible flow data (the two middle fields), the less stable areas can also arise near the center of the vortices (highlighted by the yellow arrows). This is expected because the numerical integrators, especially those with fixed step size, have trouble to compute the accurate flow paths in areas with high flow curvature, e.g., the center of a vortex.

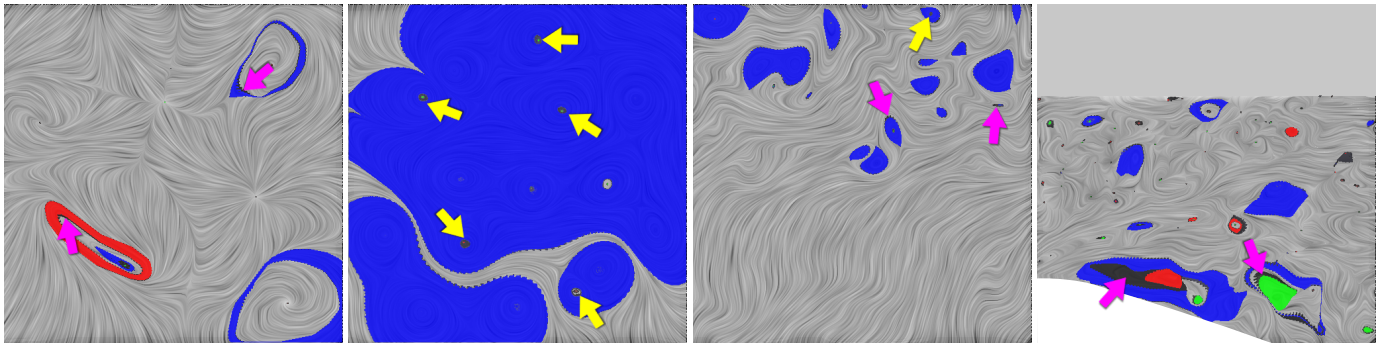


Fig. 13. The ensemble ISMDs of the PIV wind tunnel data with different amount of perturbations: 5%, 10%, 20%, and 30%. The probability  $p$  is set 0.8 and  $res = 512$ ,  $n_{stp} = 5$ ,  $n_{sd} = 16$  for all these computation.

## B. Noise in the Data

In this study, we artificially perturb the vector value at every data point to mimic the noise in the data. Assuming no prior knowledge is available, we perturb each vector by a uniform random noise. Specifically, given a vector pointing from  $\mathbf{p}$  to  $\mathbf{q}$ , a new ending point  $\mathbf{q}'$  is selected from a disk with radius  $r$  and centered at  $\mathbf{q}$ .  $r = c \times |V_{max}|$ , where  $|V_{max}|$  is the largest magnitude among the vector values in the vector field and  $c \in (0, 1]$  quantifies the amount of perturbation. Figure 14 (middle diagram) illustrates such a perturbation. With this strategy, we generate a series of perturbed vector fields from the input data for an ensemble analysis. Similar to the above study of integration error, we compute an ISMD for each perturbed vector field and record each pixel on whether it is in a Morse set or not. With the aforementioned blending process, we generate a visualization so that the pixels that are more likely on the true Morse sets will be assigned brighter colors, while the pixels with less certainty get darker colors.

We use a 2D data set obtained at the OSU wind tunnel [26] to study this. This data set represents experimentally obtained velocity components in a two dimensional plane along the centerline of a wing in a moderate Reynolds number ( $Re = 6 \times 10^4$  based on the chord length). This data set was obtained using particle image velocimetry and represents a snapshot of the velocity field with a vector resolution of 0.684mm in a total field of field of 54mm  $\times$  47mm. The wing is at a 20° angle of attack (chord line relative to flow direction) and as such experiences a leading edge separation.

Figure 15 shows the ensemble visualization of the ISMDs with different amount of perturbations being added randomly. For all these computations, we set  $res = 512$ ,  $n_{stp} = 5$ ,  $n_{sd} = 16$ , and  $N_{trials} = 20$ . From these results, we see that with more perturbations the obtained Morse sets tend to have larger less certain areas, i.e., those dark regions. Also, in most cases, those less certain areas are typically located at the boundaries of the Morse sets (also

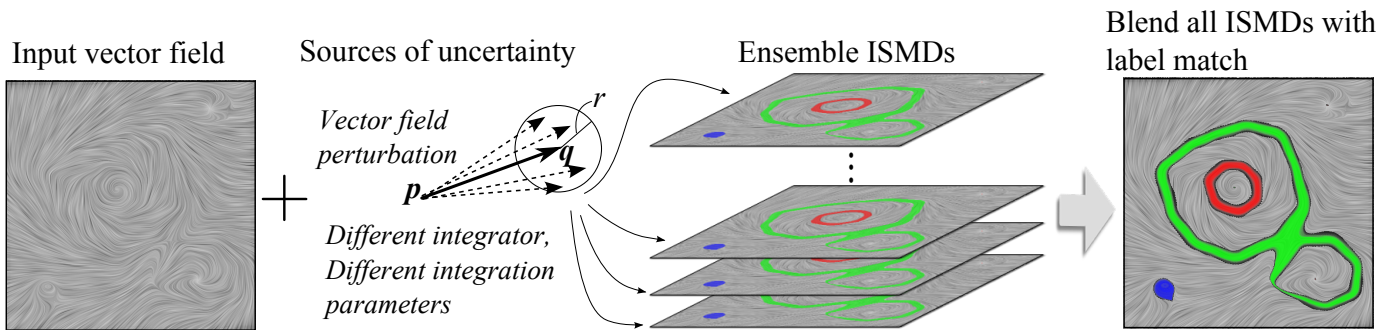


Fig. 14. Generation of the ISMD ensemble. First, each vector value of the input vector field is perturbed to generate a series of new vector fields, as described in the middle illustration. Second, ISMDs of these new vector fields are computed and recorded. Finally, this ensemble of ISMDs are blended to generate the visualization shown in the right image.

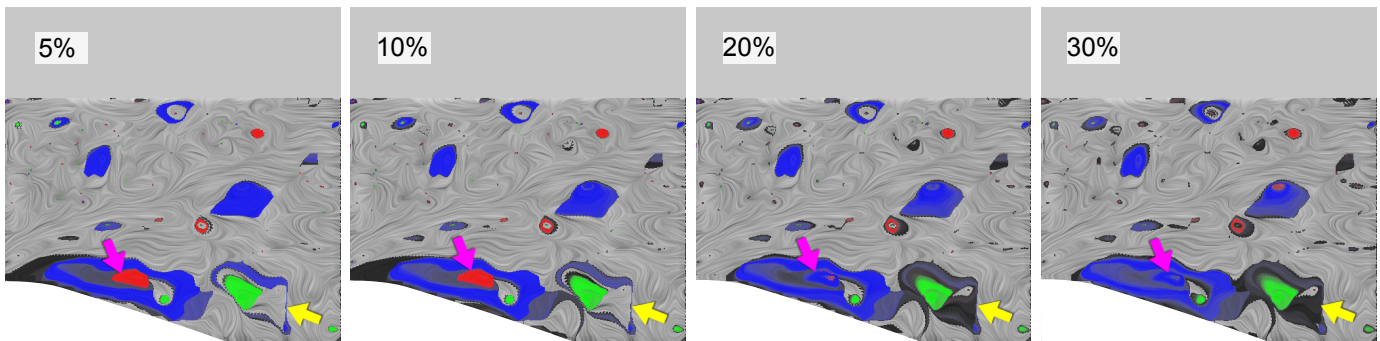


Fig. 15. The ensemble ISMDs of the PIV wind tunnel data with different amount of perturbations: 5%, 10%, 20%, and 30%. The probability  $p$  is set 0.8 and  $res = 512$ ,  $n_{stp} = 5$ ,  $n_{sd} = 16$  for all these computation.

shown in Figure 1(c)), which is expected as the points (here pixels) near the boundary of a Morse sets are most influenced by the perturbation. This can be explained by the behavior of the strongly connected components of a directed graph. As shown in Figure 16, a small perturbation may change some edges of the graph. The change of the edges at the boundary (Figure 16(a)) of the strongly connected component has larger influence than those in the interior (Figure 16(b)) of the strongly connected component.

From the results shown in Figure 15, we observe that two Morse sets are less stable under perturbation than the others. The Morse set highlighted by the yellow arrows becomes less stable, shown by the dark color, with the increasing amount of perturbation. Similarly, the Morse set highlighted by the magenta arrows changes type, i.e., from sink-like to saddle-like Morse set, with the increasing amount of perturbation. Both Morse sets are close to the wing that indicate certain vortex behavior of the air. However, what causes this instability will need in-depth investigation and the discussion with the domain experts.

## VI. CONCLUSION

In this work, we present an Image-Space Morse decomposition (ISMD). Compared to the original Morse decomposition that is computed in the object-space, the ISMD first projects the input vector field onto an image plane. Then, a directed graph,  $\mathcal{F}_I$ , is constructed based on the projected vector field defined in the image plane, where the graph nodes correspond to the individual pixels and the directed edges indicate the flow mapping relation between pixels. This is in contrast to the object-space method where the graph nodes correspond to the triangles (or other polygonal elements) of the space discretization. Next, the Morse decomposition is computed using the original method based on  $\mathcal{F}_I$ . With this modification, ISMD addresses the boundary smoothness issue of the original representation and achieves pixel-level accuracy in the visualization. In addition, with the ISMD framework, the stability of the Morse sets under different sources of uncertainty, including the error introduced in the numerical integration and the perturbation or noise in the data, can be studied and visualized, which is difficult to achieve via the original method. We have also provided a thorough study of the performance of the proposed ISMD and applied it to a number of synthetic and real-world vector field data to demonstrate its utility.



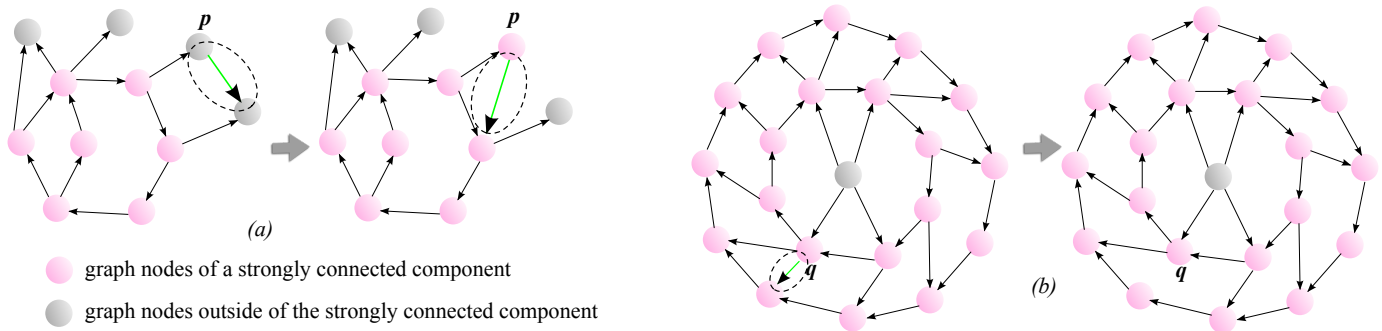


Fig. 16. Illustration of how small perturbation can affect the detected strongly connected components. (a) the perturbation causes the change of the direction of an edge (starting from a node  $p$ , highlighted as green), which makes  $p$  becomes part of a strongly connected component. (b) the perturbation removes an interior edge starting from a node  $q$  (highlighted as green). This does not change the original strongly connected component.

Similar to other image-space approaches, ISMD is also view-dependent and requires recomputation during user interaction. In addition, certain features will be missed when they are partially outside the viewport. Further, the memory consumption of the ISMD may be larger than the object-space method if the resolution of the input mesh is coarser than the image resolution. Also, the uniform sampling strategy for placing particles for the estimation of the flow map need not be as accurate as the object-space method. Therefore, instead of claiming the proposed ISMD is better than the original method, we consider it a valuable addition to the existing Morse decomposition techniques, which may provide a quick overview of the flow recurrent behavior.

In the future, we would like to extend the current ISMD framework to handle the tangential vector fields defined on curved surfaces and 3D vector fields. We would also like to combine the ISMD framework with the object-space method to further enhance the information encoding during the construction of the graph for the development of a new and unified graph-based vector field analysis framework, and continue improving the smoothness of the visual representation of the decomposition results. In addition, we will explore the utility of ISMD framework for the uncertainty analysis and visualization of vector fields, as well as the hierarchical representation of the flow structure. Finally, we plan to improve the performance of the current ISMD by studying a better sampling strategy and more compact representation of the graph.

## REFERENCES

- [1] G. Chen, K. Mischaikow, R. S. Laramee, and E. Zhang, "Efficient Morse Decompositions of Vector Fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 14, no. 4, pp. 848–862, Jul./Aug. 2008.
- [2] H. Bhatia, S. Jadhav, P.-T. Bremer, G. Chen, J. A. Levine, L. G. Nonato, and V. Pascucci, "Flow visualization with quantified spatial and temporal errors using edge maps," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, pp. 1383–1396, 2012.
- [3] J. Reininghaus, C. Lowen, and I. Hotz, "Fast combinatorial vector field topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, pp. 1433–1443, 2011.
- [4] A. Szymczak and E. Zhang, "Robust morse decompositions of piecewise constant vector fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 6, pp. 938–951, June 2012. [Online]. Available: <http://dx.doi.org/10.1109/TVCG.2011.88>
- [5] G. Chen, K. Mischaikow, R. S. Laramee, P. Pilarczyk, and E. Zhang, "Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 769–785, Jul./Aug. 2007.
- [6] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The State of the Art in Flow Visualization: Feature Extraction and Tracking," *Computer Graphics Forum*, vol. 22, no. 4, pp. 775–792, Dec. 2003. [Online]. Available: <http://cs.swan.ac.uk/~csbob/research/>
- [7] R. S. Laramee, H. Hauser, H. Doleisch, F. H. Post, B. Vrolijk, and D. Weiskopf, "The State of the Art in Flow Visualization: Dense and Texture-Based Techniques," *Computer Graphics Forum*, vol. 23, no. 2, pp. 203–221, June 2004. [Online]. Available: <http://www.VRVis.at/ar3/pr2/star/>
- [8] T. Salzbrunn, T. Wischgoll, H. Jänicke, and G. Scheuermann, "The state of the art in flow visualization: Partition-based techniques," in *In Simulation and Visualization 2008 Proceedings*, H. Hauser, S. Strassburger, and H. Theisel, Eds. SCS Publishing House, 2008, pp. 75–92.
- [9] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, , and M. Chen, "Over two decades of integration-based, geometric flow visualization," in *proceeding of EUROGRAPHICS 2009, State of the Art Reports*, 2009, pp. 73–92.
- [10] A. Pobitzer, R. Peikert, R. Fuchs, B. Schindler, A. Kuhn, H. Theisel, K. Matkovic, and H. Hauser, "The state of the art in topology-based visualization of unsteady flow," *Computer Graphics Forum*, vol. 30, no. 6, pp. 1789–1811, September 2011. [Online]. Available: <http://dx.doi.org/10.1111/j.1467-8659.2011.01901.x>

- [11] M. Edumunds, R. S. Laramée, G. Chen, N. Max, E. Zhang, and C. Ware, "Surface based flow visualization," *Computers & Graphics, to appear*.
- [12] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets." *IEEE Computer*, vol. 22, no. 8, pp. 27–36, 1989.
- [13] G. Scheuermann, B. Hamann, K. Joy, and W. Kollmann, "Visualizing local topology," *Journal of Electronic Imaging*, vol. 9, no. 4, 2000.
- [14] G. Scheuermann, H. Krüger, M. Menzel, and A. P. Rockwood, "Visualizing nonlinear vector field topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 2, pp. 109–116, 1998.
- [15] T. Wischgoll and G. Scheuermann, "Detection and Visualization of Closed Streamlines in Planar Fields," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 165–172, 2001.
- [16] —, "Locating Closed Streamlines in 3D Vector Fields," in *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym 02)*, May 2002, pp. 227–280.
- [17] H. Theisel, T. Weinkauff, H.-P. Seidel, and H. Seidel, "Grid-Independent Detection of Closed Stream Lines in 2D Vector Fields," in *Proceedings of the Conference on Vision, Modeling and Visualization 2004 (VMV 04)*, Nov. 2004, pp. 421–428.
- [18] H. Theisel, T. Weinkauff, H.-C. Hege, and H.-P. Seidel, "Saddle connectors - an approach to visualizing the topological skeleton of complex 3d vector fields," in *VIS '03: Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. Washington, DC, USA: IEEE Computer Society, 2003, p. 30.
- [19] T. Weinkauff, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel, "Extracting higher order critical points and topological simplification of 3d vector fields," in *Visualization, 2005. VIS 05. IEEE*, Oct. 2005, pp. 559–566.
- [20] W. Reich, S. D., H. C., W. A., G. Chen, and G. Scheuermann, "Combinatorial vector field topology in three dimensions," in *Topology-based Methods in Visualization*, ser. Mathematics and Visualization, R. Peikert, H. H., , and F. R., Eds. Springer, 2011, pp. 47–62.
- [21] G. Chen, Q. Deng, A. Szymczak, R. S. Laramée, and E. Zhang, "Morse set classification and hierarchical refinement using conley index," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 5, pp. 767–782, 2012.
- [22] C. Conley, *Isolated Invariant Sets and Morse Index*, ser. CBMS Lecture Notes. AMS Providence, RI, 1978, vol. 38.
- [23] W. D. Kalies, K. Mischaikow, and R. C. A. M. VanderVorst, "An algorithmic approach to chain recurrence," *Found. Comput. Math.*, vol. 5, no. 4, pp. 409–449, 2005.
- [24] S. Shadden, F. Lekien, and J. Marsden, "Definition and properties of lagrangian coherent structures from finite-time lyapunov exponents in two-dimensional aperiodic flows," *Physica D*, vol. 212, no. 3–4, pp. 271–304, 2005.
- [25] S. S. Barakat and X. Tricoche, "Adaptive refinement of the flow map using sparse samples," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 12, pp. 2753–2762, 2013.
- [26] G. Chen, Z. Lin, D. Morse, S. Snider, S. Apte, J. Liburdy, and E. Zhang, "Multiscale feature detection in unsteady separated flows." 2008.
- [27] J. H. Chen, E. R. Hawkes, R. Sankaran, S. D. Mason, and H. G. Im, "Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: I. fundamental analysis and diagnostics," *Combustion and flame*, vol. 145, no. 1, pp. 128–144, 2006.
- [28] P. Skraba, B. Wang, G. Chen, and P. Rosen, "2d vector field simplification based on robustness," in *Pacific Visualization Symposium (PacificVis), 2014 IEEE*. IEEE, 2014, pp. 49–56.
- [29] J. Huang, Z. Pan, G. Chen, W. Chen, and H. Bao, "Image-space texture-based output-coherent surface flow visualization," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 19, no. 9, pp. 1476–1487, 2013.
- [30] W. Reich and G. Scheuermann, "Analysis of streamline separation at infinity using time-discrete markov chains," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2140–2148, 2012.
- [31] L. Xu and H.-W. Shen, "Flow web: A graph based user interface for 3d flow field exploration," in *Proceeding of IS&T/SPIE Visualization and Data, VDA'10*, 2010.
- [32] V. Kwatra, D. Adalsteinsson, T. Kim, N. Kwatra, M. Carlson, and M. C. Lin, "Texturing fluids," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 13, no. 5, pp. 939–952, 2007.