

# Efficient Morse Decompositions of Vector Fields

Category: Research

## Abstract—

Vector field analysis plays a crucial role in many engineering applications, such as weather prediction, tsunami and hurricane study, and airplane and automotive design. Existing vector field analysis techniques focus on individual trajectories such as fixed points, periodic orbits and separatrices which are sensitive to noise and errors introduced by simulation and interpolation. This can make such vector field analysis unsuitable for rigorous interpretations. In this paper, we advocate the use of Morse decompositions, which are robust with respect to perturbations, to encode the topological structures of the vector field in the form of a directed graph, called a Morse decomposition connection graph (MCG). While an MCG exists for every vector field, it need not be unique. We develop the idea of a  $\tau$  map, which decouples the MCG construction process and the configuration of the underlying mesh. This, in general, results in finer MCGs than mesh-dependent approaches.

To compute MCGs effectively, we present an adaptive approach in constructing better approximations of the images of triangles in the meshes used for simulation. These techniques result in fast and efficient MCG construction. We demonstrate the efficacy of our technique on various examples in planar fields and on surfaces including engine simulation data.

**Index Terms**—Vector field topology, Morse decomposition, multi-valued map, connection graph

## 1 INTRODUCTION

Detecting patterns of fluid flow, i.e., feature-extraction, is an important topic in vector field visualization with many applications [17]. Extracting and visualizing vector field topology has important applications in Computational Fluid Dynamics (CFD) [13], weather prediction, tsunami and hurricane study, and airplane design. In particular, fixed points, recirculation zones or periodic orbits can be both desirable [2] or undesirable [12] in engine simulation, depending on their location.

Past work, such as the extraction of the vector field topological skeleton [7, 8] and *Entity Connection Graph* (ECG) [2], which is a superset of the topological skeleton with periodic orbits being considered, defines vector field topology as fixed points, periodic orbits and the separatrices connecting them. However, analysis and visualization of vector field topology based on individual trajectories can raise questions with respect to interpretation. The discrete nature of fluid flow data poses several challenges. Firstly, data samples are given at discrete locations, e.g., only at cell vertices or cell centers. Interpolation schemes are then used in order to reconstruct the vector field at locations between the given samples. Also, the given data samples themselves are numerical approximations, e.g., approximate solutions to a set of partial differential equations. Thirdly, the given flow data are often only a linear approximation of the underlying dynamics. Finally, the visualization algorithms themselves, e.g., streamline integrators, have a certain amount of inherent error associated with them. In short, how can we be sure that what we see is authentic when extracting and visualizing the topological skeleton of flow field? Could the error inherent to multiple numerical approximations produce misleading information? Figures 1 and 2 provide examples in which proper interpretation can be difficult when performing analysis based on individual trajectories.

In Figure 1, a number of vector fields have similar appearances (top) despite having rather different vector field topologies (bottom). How can we be sure there is no periodic orbit, or only one periodic orbit in the flow? Figure 2 (left) shows an example in which one of the separatrices starting from the saddle in the bottom of the flow connects to one of the sources in the upper right corner of the flow. Due to the numerical errors of the streamline integrator, the separatrix can connect to either source. Figure 2 (right) shows a field containing one saddle and two center-like fixed points. A small perturbation can result in the incorrect extraction of these fixed points. If they are detected as highly curled foci, the separatrices starting from the center saddle are supposed to reach them in a certain amount of time (or distance). However, this may not happen due to the limit of the maximum distance of tracing allowed.

In order to address these important challenges, we present a novel, robust algorithm for the representation, extraction and visualization of flow topology. In particular, we incorporate fundamental ideas from dynamical systems, the notion of *Morse decompositions* [3, 11], which can be represented by an acyclic directed graph, given the spatial discretization of the phase space (a polygonal domain). The nodes in the graph, referred to as *Morse sets*, contain all the recurrent dynamics of the vector field. The edges indicate how the flow moves from one polygon to the next. This graph which encodes the global dynamics is called the *Morse decomposition connection graph* (MCG). An MCG is stable with respect to perturbation, which makes it possible to provide rigorous interpretation of its meanings. In contrast, trajectory-based topology representations are not.

Chen et al [2] construct Morse decompositions based on the behaviors of the vector field along edges of the triangles. Because the triangulation is not adapted to the vector field, this can result in coarse Morse sets (Figure 3, top). In this paper we exploit a temporal discretization, which we refer to as a  $\tau$ -map, by integrating the vector field for a finite amount of time, though we allow this time to be spatially dependent. This results in finer MCGs (Figure 3, bottom) from which we can extract additional information about the topological structure of the dynamics of the flow.

This approach yields the following advantages. First, rather than extracting discrete, singular streamline-based segmentations of the flow, this approach detects *regions* of segmented flow that are less misleading with respect to their interpretation. Second, more rigorous statements about the dynamics of the underlying flow can be made because Morse decomposition accounts for the numerical error inherent in the vector field data [11]. In other words, a well defined error,  $\epsilon > 0$ , can be bounded and included into the map of the flow domain. We point out that addressing such uncertainty in visualization was identified as one of the most important future challenges by Johnson [9].

The key challenges with this approach are choosing an appropriate spatial discretization of the flow and constructing a good multi-valued map, which is the discrete outer approximation of a  $\tau$  map, from which we extract the Morse sets—two topics addressed in this paper. The work presented here yields the following benefits and contributions:

- A sound, theoretical framework based on Morse decompositions from which more certain and rigorous statements can be made with respect to the extraction of flow topology.
- A means to get finer Morse decompositions of a given vector field using the idea of a  $\tau$  map.
- A heuristic implementation on the efficient construction of the multi-valued maps and consequent fine Morse decompositions.

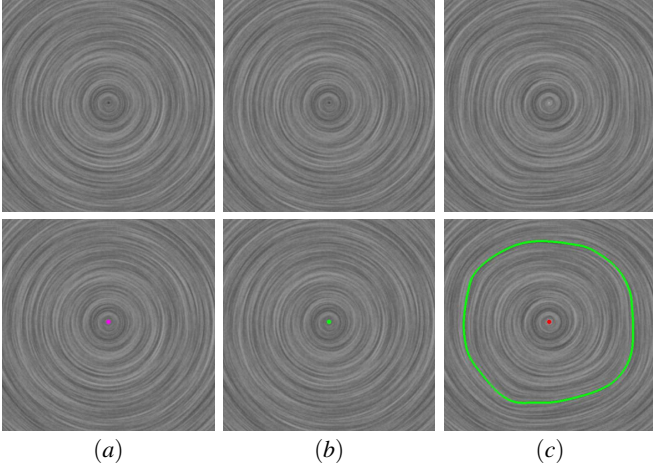


Fig. 1. Noise and piecewise linear representations of vector fields can cause difficulties when extracting the nodes in vector field topology, such as fixed points and periodic orbits. The three fields look alike (top), yet their vector field topologies are very different (bottom). (a) a center; (b) a repelling focus; (c) a fixed point and a periodic orbit. We make use of the periodic orbit extraction algorithm of Chen et al. [2] to detect periodic orbits in the example fields shown in this paper.

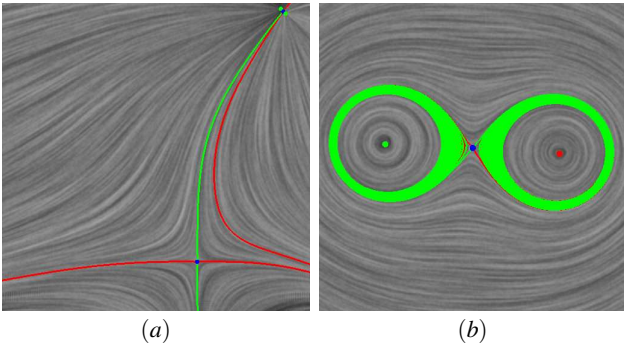


Fig. 2. Noise, piecewise linear approximation and discretization can cause difficulties when extracting the connections between nodes in a vector field topology, such as separatrices. In (a), it is not clear which of the three fixed points in the top is connected to the bottom saddle. In (b), it is difficult to determine whether the saddle is connected to the fixed points even after a long integration.

- The application of the proposed topological analysis technique to both analytical planar data and application-oriented data sets including engine simulation data from CFD on 3D surfaces.

The rest of this paper is organized as follows. Section 2 provides a brief review of related work on vector field topology. Section 3 introduces the Morse decompositions and multi-valued maps. Section 4 describes an algorithm to compute the multi-valued maps. Section 5 presents the algorithm for MCG construction. Section 6 shows the utility of our approach to the engine simulation data followed by a summary and discussion of possible future work in Section 7.

## 2 RELATED WORK

Helman and Hesslink introduced the visualization community to the notion of flow topology in 1989 [7] and 1991 [8], respectively. Their analysis included the detection, classification, and visualization of fixed points in planar flows. They applied their algorithms to both steady-state and unsteady flow. They represented time as a third spatial dimension for the case of time-dependent, planar flow.

**Extraction of Fixed Points:** Most fixed point detection algorithms are based on piecewise linear or bilinear approximation. Tricocche et al. [25] and Polthier and Preuß [16] give efficient methods to locate fixed points in a vector field. These methods do not properly represent local topology if nonlinear behavior is present. Scheuermann

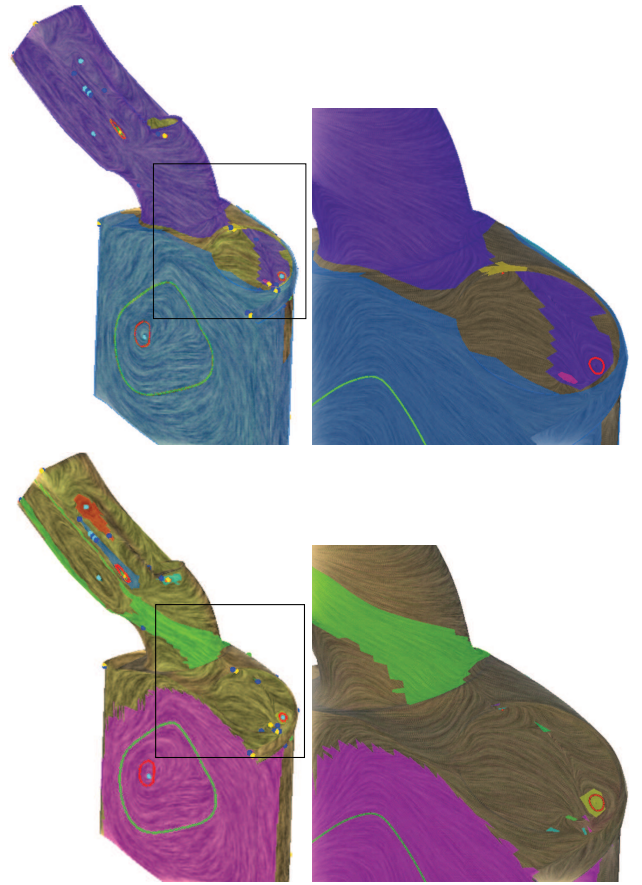


Fig. 3. A comparison of the obtained Morse sets of gas engine simulation data with the geometry-based map (top) and a multi-valued map with  $d_\tau = 0.5$  (bottom), respectively, where  $d_\tau$  is the spatial discretization of the trajectory integral. The colored regions indicate the Morse sets. The triangles in the same Morse set are displayed in the same color. The gray shaded regions indicate the gradient-like flow. Observe that using the  $\tau$  map results in smaller isolated regions in which we can further extract and identify topological structures of the flow.

et al. [18, 20] choose a polynomial approximation in areas with non-linear behavior and apply a suitable visualization—streamlines seeded at the fixed points with additional annotations.

**Extraction of Periodic Orbits:** Wischgoll and Scheuermann [28] present an algorithm for detecting periodic orbits in planar flows, which are of interest because they may indicate regions of recirculating flow. It is based on monitoring streamlines as they enter, exit, and re-enter cells of the vector field domain. We urge the reader to use caution when interpreting the visualization results. This is because a spatial dimension inherent to the applied domain has been left out of the analysis. Wischgoll and Scheuermann [29] extend their previous work [28] of detecting periodic orbits to 3D vector fields. The algorithm is based on preventing infinite cycling during streamline integration. This approach has also been extended to time-dependent flows by Wischgoll et al. [30]. At each time step, periodic orbits are extracted. Afterward, a time-dependent correspondence between individual streamlines is computed. Theisel et al. [23] present an alternative approach to compute periodic orbits. A 2D vector field is transformed into a 3D vector field, which can be done by representing time as a third spatial dimension. Then streamsurfaces are seeded in the 3D domain. Finally, periodic orbits are detected by intersecting streamsurfaces. The difference to previous work [28] is that this approach avoids mesh-based dependency, e.g., examining and testing individual mesh polygons. Chen et al. [2] propose a novel periodic orbit extraction algorithm based on Morse decomposition, which can detect periodic orbits that do not connect with any fixed points efficiently.

See Laramée et al. for a complete overview of topology-based flow visualization research [13]. In general, previous topology extraction techniques rely on linear algebra in order to detect topological features of vector fields without regard to the numerical uncertainty described in the introduction.

### 3 BACKGROUND

In this section, we present a compact summary of the dynamical systems theory upon which our work is built.

#### 3.1 Morse Decompositions

We are interested in describing the topological structures of the flow generated by a vector field  $\dot{x} = f(x)$  defined on a triangular surface  $X$ .

However, the information we are given consists of a finite set of vectors

$$\{f_d(v_i) \mid v_i \text{ a vertex of } X\} \quad (1)$$

obtained either by a large numerical simulation or from experimental data. This means that at best we can assume that we have a uniform bound on the errors of the observed vector field versus the true vector field, that is for each  $v_i$ ,

$$\|f(v_i) - f_d(v_i)\| \leq \varepsilon. \quad (2)$$

In addition, since we are only given the data (Eq.1) we extend  $f_d$  to a vector field on  $X$  by some means of interpolation (typically linear interpolation). Assuming that  $f$  is well approximated by  $f_d$  it is reasonable to assume that the bounds of (Eq.2) are global, that is  $\|f(x) - f_d(x)\| \leq \varepsilon$  for all  $x \in X$ .

The easiest way to encode the aforementioned information is to consider a family of vector fields  $F$  defined on the surface  $X$  and parameterized by some abstract parameter space  $\Lambda$ . We assume that for each  $\lambda \in \Lambda$ , the vector field  $\dot{x} = F(x, \lambda)$  gives rise to a flow  $\varphi_\lambda : \mathbb{R} \times X \rightarrow X$ .

In this setting we assume that there exists parameter values  $\lambda_0, \lambda_1 \in \Lambda$  such that  $f(x) = F(x, \lambda_0)$  and  $f_d(x) = F(x, \lambda_1)$ . Bifurcation theory tells us that even if  $\lambda_0 \approx \lambda_1$ , the orbits, i.e. fixed points, periodic orbits, separatrices, of  $\varphi_{\lambda_0}$  and  $\varphi_{\lambda_1}$  need not agree. The implication is that computing such orbits for the vector field  $f_d$  does not imply that these orbits exist for the true vector field  $f$  (Figure 1 (c)).

This leads us to weaken the topological structures which we use to classify the dynamics. A *Morse decomposition* of  $X$  for a flow  $\varphi_\lambda$  is a finite collection of disjoint compact invariant sets, called *Morse sets* [11]

$$\mathbf{M}(X, \varphi) := \{M_\lambda(p) \mid p \in (\mathcal{P}_\lambda, \succ_\lambda)\},$$

where  $\succ_\lambda$  is a strict partial order on the indexing set  $\mathcal{P}_\lambda$ , such that for every  $x \in X \setminus \cup_{p \in \mathcal{P}_\lambda} M_\lambda(p)$  there exist indices  $p \succ q$  such that

$$\omega(x) \subset M_\lambda(q) \quad \text{and} \quad \alpha(x) \subset M_\lambda(p).$$

It is easy to check that any structures that are associated with recurrent dynamics of  $\varphi_\lambda$ , i.e. fixed points, periodic orbits, chaotic dynamics, must lie in the Morse sets. The dynamics outside the Morse sets is gradient-like. Morse decompositions of invariant sets always exist, though they may be trivial, i.e. consisting of a single Morse set  $X$ .

Observe that since  $\mathcal{P}_\lambda$  is a strictly partially ordered set a Morse decomposition can be represented as an acyclic directed graph. The nodes of the graph correspond to the Morse sets and the edges of the graph are the minimal order relations which through transitivity generate  $\succ_\lambda$ . This graph is called the Morse decomposition connection graph and denoted by  $MCG_\lambda$ .

#### 3.2 Computing Morse Decompositions

We now turn to the question of how to compute MCGs. The first step is to move from the continuous time of a flow to discrete time of a map. This leads to the following definition.

**Definition 3.1** Let  $\tau : X \rightarrow (0, \infty)$  be a continuous map. A  $\tau$ -time discretization of the flow  $\varphi$  is a map  $f_\tau : X \rightarrow X$  defined by

$$f_\tau(x) := \varphi(\tau(x), x).$$

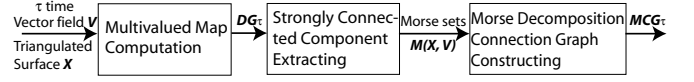


Fig. 4. The pipeline of computing the Morse decomposition of a vector field. The input is a vector field  $V$  defined on a triangle mesh  $X$ . The output is the Morse decomposition of the vector field represented as the Morse decomposition connection graph (MCG).

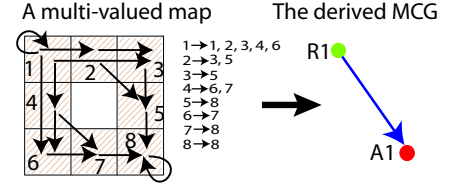


Fig. 5. This figure illustrates an example of a multi-valued map (left) and the derived MCG (right). The flow is defined in the shaded region only. The edges in the multi-valued map demonstrate the mapping relations of the polygons. The multi-valued map forms a directed graph (not shown in this figure), which we use to compute the Morse decomposition and construct the MCG (right).

Kalies, Mischaikow, and Mrozek (work in progress) have proven that  $S$  is an isolated invariant set for  $\varphi$  if and only if  $S$  is an isolated invariant set for a  $\tau$ -time discretization  $f_\tau$ . Thus, finding Morse decompositions for the flow  $\varphi$  is equivalent to finding Morse decompositions for  $f_\tau$ .

The fact that  $X$  is a triangulated surface provides us with an appropriate discretization in space. Let  $\mathcal{X}$  be the triangulation of  $X$ . We will approximate  $f_\tau$  using a *combinatorial multi-valued map*  $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$ , that is a map such that for each triangle  $T \in \mathcal{X}$ , its image is a set of triangles, i.e.  $\mathcal{F}(T) \subset \mathcal{X}$ . From the point of view of computation it is useful to view  $\mathcal{F}$  as a directed graph, we denote the equivalent directed graph by  $DG_\tau$ . The vertices of  $DG_\tau$  are the triangles and the edges indicate the images of the triangles. Notice while  $DG_\tau$  is not the same as the Morse decompositions, despite the fact that both are directed graphs (Figure 5). In fact, Morse decomposition is derived from  $DG_\tau$ .

The correct notion of approximation is given by the following definition. Consider  $f_\tau : X \rightarrow X$ . The combinatorial multi-valued map  $\mathcal{F} : \mathcal{X} \rightrightarrows \mathcal{X}$  is an *outer approximation* of  $f_\tau$  if

$$f_\tau(T) \subset \text{int}(|\mathcal{F}(T)|)$$

for every  $T \in \mathcal{X}$  where  $|\mathcal{F}(T)| := \cup_{R \in \mathcal{F}(T)} R$ .

Observe that the definition of an outer approximation requires a lower bound on the set of triangles in  $\mathcal{F}(T)$ , but not an upper bound. In general larger images of  $\mathcal{F}$  are easier to compute. For example, one can obtain an outer approximation, by declaring  $\mathcal{F}(T) = \mathcal{X}$  for all  $T \in \mathcal{X}$ . However, the larger the image the poorer the approximation of the dynamical system of interest,  $f_\tau$ .

Using this machinery we can describe the pipeline for our method for computing an MCG. First, we choose a  $\tau$ -map,  $f_\tau$ . Second, we compute a combinatorial multi-valued map  $\mathcal{F}$  which is an outer approximation of  $f_\tau$  to get  $DG_\tau$ . Next, we extract the strongly connected components of  $DG_\tau$ . It is shown in [1] that the union of the triangles in the strongly connected components form isolating neighborhoods for Morse sets. Finally, we form an acyclic directed graph, the  $MCG_\tau$  whose vertices consist of the strongly connected components of  $DG_\tau$  and whose edges inherit the path ordering from  $DG_\tau$ . Figure 4 illustrates the pipeline.

Figure 6 shows an example MCG from an analytical vector field generated by Eq. 3.

$$V(x, y) = \begin{pmatrix} y \\ -x + y \cos(x) \end{pmatrix} \quad (3)$$

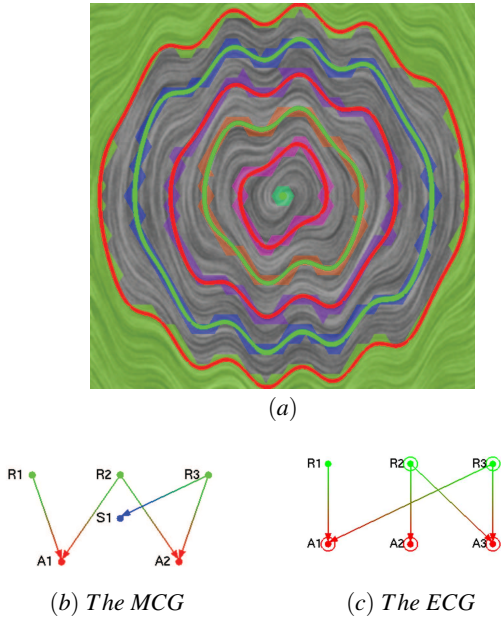


Fig. 6. This figure illustrates the Morse decomposition of an analytical vector field defined by Eq.3 over the region  $\{(x,y)|\max(|x|,|y|) < 11\pi\}$  (a). There is one source in the region enclosed by five periodic orbits [31]. The colored regions indicate the isolating neighborhood of the extracted Morse sets. Different colored regions refer to different Morse sets. The gray shaded region indicate the regions with gradient-like flow. (b) shows the MCG derived from a multi-valued map of this field with  $d_\tau = 40$ . The nodes of the MCGs are the isolating neighborhoods of the Morse sets, which are triangulated regions. The green dots indicate the *source Morse sets*, red dots refer to *sink Morse sets* and blue dots represent *saddle Morse sets*. (c) shows the ECG of the vector field. The nodes of the ECG consists of the individual fixed points or periodic orbits.

The definition of an outer approximation and the fact that the triangles in the strongly connected components of  $\mathcal{F}$  form isolating neighborhoods for the Morse sets demonstrate why the MCG remains constant under small perturbations of the vector field. Since  $f_\tau$  is a continuous map and each triangle  $T$  is compact, the image  $f_\tau(T)$  is a compact set. If  $\mathcal{F}$  is an outer approximation, then by definition  $f_\tau(T)$  is contained in the interior of the set  $|\mathcal{F}(T)|$ . Thus, this property will also hold for any sufficiently small perturbation of  $f_\tau$ , which means that given a  $DG_\tau$  for  $f_\tau$  we have the same  $DG_\tau$  for any sufficiently small perturbation of  $f_\tau$ . Of course, we can go one step further and insist that an  $\varepsilon$ -neighborhood of  $f_\tau(T)$  be contained in  $|\mathcal{F}(T)|$ . We will in general get a coarser  $DG_\tau$ , but the resulting Morse decompositions will be valid for any vector field whose  $\tau$  map lies within  $\varepsilon$  of  $f_\tau$ .

Among the modula of the pipeline, computing strongly connected components of a directed graph can be implemented using a standard graph algorithm [4], and the MCG construction can be implemented using the algorithm proposed in [10]. We will describe how to construct the multi-valued map efficiently in Section 4.

#### 4 COMPUTING THE MULTI-VALUED MAP

We now describe the computational model of our system. In this model, the underlying domain is represented by a triangular mesh. Vector values are defined at the vertices, and interpolation is used to obtain values on the edges and inside the triangles. For the planar case, we use the popular piecewise linear interpolation method of Tricoch et al. [25]. On curved surfaces, we borrow the interpolation scheme of Zhang et al. [32] which guarantees vector field continuity across the vertices and edges of the mesh. These interpolation schemes support efficient flow analysis operations on both planes and surfaces.

A multi-valued map encodes the dynamic behavior of a vector field. Accurately constructing the combinatorial multi-valued map, i.e. a di-

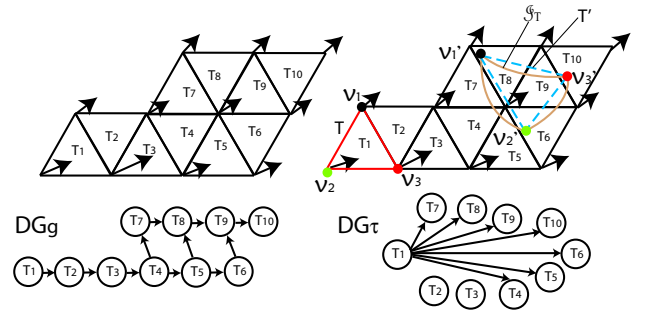


Fig. 7. This figure illustrates the difference between two MCGs that are constructed either based on behaviors on the edges (left) or according to a multi-valued map (right). Each node corresponds to a triangle of the mesh. The left column shows the construction of  $DG_g$  using the geometry-based map. The right column shows the construction of  $DG_\tau$  of a multi-valued map with  $\tau > 0$ . The red triangle  $T = T_1$  is the starting triangle, the light brown curved closure is the real image of  $T$ , the blue dash triangle is the approximation of the real image, the gray dash lines show the mapping of the vertices of  $T$ .

rected graph  $DG_\tau$ , of a vector field is crucial for computing the Morse decomposition of the vector field.

Chen et al. [2] describe an approach to compute the  $DG_\tau$  of a given vector field  $V$  by considering the flow behavior across the edges of each triangle and add the directed edges to and from neighboring triangles accordingly (Figure 7, left). We refer to this approach as the geometry-based map, which differs from a multi-valued map based on a given  $\tau$  map. In the remainder of this paper, we will refer to the directed graph obtained using this method as  $DG_g$ .

$DG_g$  often results in a rather coarse outer approximation of the underlying dynamics, i.e., Morse sets that contain multiple fixed points and periodic orbits or no structures at all. For instance, the Morse sets extracted using  $DG_g$  of the gas engine simulation data set are shown in Figure 3. The colored regions indicate the isolating neighborhoods of the Morse sets. Different color regions indicate different Morse sets. The gray shaded regions indicate the regions of gradient-like flow. We observe that with the geometry-based method (first row) there are two Morse sets in the cylinder and intake port of the engine covering a large portion of the surface geometry. This makes subsequent analysis, such as ECG computation [2], more difficult and computationally expensive.

In order to obtain finer Morse sets, we make use of the idea of multi-valued maps introduced in Section 3.2 to compute the directed edges of the  $DG_\tau$ . A three-step procedure of this idea is described as follows (Figure 7, right).

- First, trace from random samples of each triangle  $T$  of the surface  $X$  with  $\tau$  time.
- Second, reconstruct the advected triangle  $\mathcal{S}_T$  based on the new positions of the samples.  $\mathcal{S}_T$  will intersect with a set of triangles  $T_i$  of the surface  $X$ .
- Third, add the directed edges from the node of  $T$  to the nodes of  $T_i$  in the directed graph  $DG_\tau$ .

To implement the idea of a multi-valued map, we extend the time discretization  $\tau$  into a spatial discretization  $d_\tau$ . That is, for each sample of the triangle  $T$  in  $X$ , we keep track of the integral length of the sample following the flow until the integral length is larger than  $d_\tau$ . Due to the continuity of the vector fields in 2D planes and on 3D surfaces discussed in this paper, using the constant  $d_\tau$  to perform the tracing is equivalent to a continuous  $\tau$  inside the vector field.

##### 4.1 $\tau$ Map Guided Outer Approximation

The most difficult step of the three step procedure is how to compute the approximate advected triangle,  $\mathcal{S}_T$ , of a triangle  $T \subset X$  and obtain the directed edges associated with the node representing  $T$  in the  $DG_\tau$

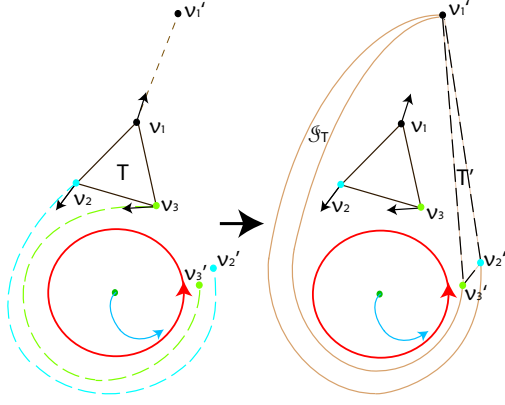


Fig. 8. This figure demonstrates a case of a distorted image of a triangle  $T$  consisting of  $v_1, v_2$  and  $v_3$  using a large  $d_\tau$ . The red closed curve represents a periodic orbit. The colored dash lines in the left figure show the trajectories of the two vertices. The light brown curved closure in the right figure shows the real image  $\mathcal{S}_T$  of the original triangle  $T$ , while the dash triangle  $T'$  is the approximate image.

efficiently. From Section 3.2, it means to compute the outer approximation of  $\mathcal{S}_T$  under  $X$ . We now describe several methods to solve this problem.

The first approach is to sample a number of points inside a triangle  $T$  and keep track of their images under the flow after a distance  $d_\tau$ . The set of the triangles containing the ending positions of these samples form an approximation of the advected triangle  $\mathcal{S}_T$ . However, it is difficult to determine the number and location of the samples necessary for each triangle to obtain an accurate approximation of the image of the triangle with respect to the flow. And it is rather computationally expensive.

The second method is derived in Section 3.2. We approximate the convex hull of the image  $\mathcal{S}_T$  through reconstructing an advected triangle  $T'$  using the three advected vertices of the original triangle  $T$ . More specifically, we trace the three vertices  $v_1, v_2, v_3$  of  $T$  with  $d_\tau$  distance and obtain  $v'_1, v'_2, v'_3$ . Then, we compute the approximate  $\mathcal{S}_T$  by connecting  $v'_1, v'_2, v'_3$  sequentially to form  $T'$ .  $T'$  will intersect with a set of triangles  $T_i$ , which is the outer approximation of  $\mathcal{S}_T$ . Figure 7 (right) illustrates this approach. The blue dash triangle is the approximate convex hull  $T'$ , while the light brown curved closure is the actual advected triangle  $\mathcal{S}_T$ . Although this method can avoid placing dense samples inside a triangle, it poses challenges. First, the approximation using the convex hull may lead to smaller image than desired. For instance, in Figure 7 (right),  $\mathcal{S}_T$  intersects with triangle  $T_4, T_5, T_6$ , but  $T'$  does not. Second, it may fail to produce an outer approximation when using a large  $\tau$  under a highly curled field. In Figure 8, the vertices ( $v_1, v_2$  and  $v_3$ ) of a triangle have been advected according to the underlying flow whose images are  $v'_1, v'_2$  and  $v'_3$ , respectively. Using the convex hull, we will obtain a triangle  $T'$  (the black dash triangle) while the real image should be the light brown curved closure  $\mathcal{S}_T$ . In this case, the approach leads to an incorrect approximation. Adding more samples on the boundary edges may solve these two problems, but will also increase the computational time. Furthermore, when applied to surface vector fields, this method requires geodesic maps to get the closure of  $T'$ , which can be computationally expensive.

#### 4.2 An Adaptive Approach for Computing the Outer Approximation

In this section, we describe a method that can obtain enough information about the image  $\mathcal{S}_T$  through the tracing of vertices without having to compute the convex hull. We start the discussion of our approach using the following observations. First, we trace the three vertices  $v_1, v_2, v_3$  of a triangle  $T$  under the vector field  $V$ . After  $d_\tau$  distance, we obtain the three advected vertices falling in three triangles  $T_{i_1}, T_{i_2}, T_{i_3}$ , respectively. They form an approximation of the image  $\mathcal{S}_T$  of  $T$ . If

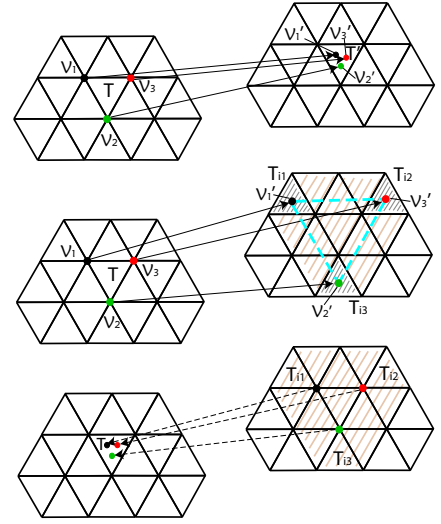


Fig. 9. This figure shows the idea of backward and forward tracing with  $d_\tau$ . The first row shows the tracing of three vertices of the original triangle  $T$  follow the vector field  $V$  and ending at a triangle  $T'$ . The second row shows that the three advected vertices fall in triangles  $T_{i_1}, T_{i_2}$  and  $T_{i_3}$ , respectively. The approximate advected triangle is the dash blue triangle. The outer approximation of the image of  $T$  is the set of the shaded triangles. The third row shows the tracing of three vertices of a shaded triangle following the vector field  $-V$ . The advected vertices fall in  $T$ , which will return the directed edges from  $T$  to all shaded triangles according to our approach. The solid black lines in the figure of the first and the second row indicate the direction of the forward mapping, while the dash ones in the figure of the third row represent the direction of the backward mapping.

$V$  is convergent,  $T_{i_1}, T_{i_2}, T_{i_3}$  either are the same triangle (Figure 9, the first row) or share common edges. Therefore, they are a good approximation of  $\mathcal{S}_T$  (Figure 9,  $T'$  in the figure of the first row). But if  $V$  is divergent, the set of  $T_{i_1}, T_{i_2}, T_{i_3}$  may not give rise of a continuous set of triangles of  $\mathcal{S}_T$  (Figure 9, second row). Second, if we reverse the vector field  $V$  and obtain  $-V$ , then the convergent flow becomes divergent and vice versa. Therefore, in Figure 9, if we trace the vertices inside the approximate closure (the shaded triangles in the second row) of the image  $\mathcal{S}_T$  of  $T$  following  $-V$ , they will gradually reach  $T$  after a distance  $d_\tau$ .

These observations motivate us to introduce the backward mapping as the complement of forward mapping when computing the image of a triangle. Our approach then can be described as follows. We first trace each vertex  $v$  of a triangle  $T$  backward with  $d_\tau$ , if it falls in triangle  $T_i$ , we add the edges from  $T_i$  to the triangles of the one-ring neighbors of  $v$  in the directed graph  $DG_\tau$ . Secondly, we trace each vertex  $v$  of  $T$  following the flow and find the triangle  $T'_i$  containing the advected vertex of  $v$ . Then, we add the edges from the triangles of the one-ring neighbors of  $v$  to  $T'_i$ . If the advected vertex is on an edge, we add edges from (backward) or to (forward) the two triangles sharing this edge. If the advected vertex ends at a vertex  $v'$ , we add edges from (backward) or to (forward) the triangles of the one-ring neighbors of  $v'$ .

Although this approach tends to work well for flows when little stretching is introduced, it may have difficulty when large stretching is present. For instance, consider a flow containing an attracting periodic orbit. A triangle in this flow may be mapped to a narrow tube in the triangle strip containing the periodic orbit. Figure 10 (left) demonstrates this with an example. In Figure 10 (left) the forward mapping will map a triangle  $T$  to  $\mathcal{S}_T$  contained by a triangle strip  $S_T$ . According to the aforementioned procedure, the backward tracing will start from the vertices of  $S_T$ . Observe that no advected vertices will fall in  $T$  due

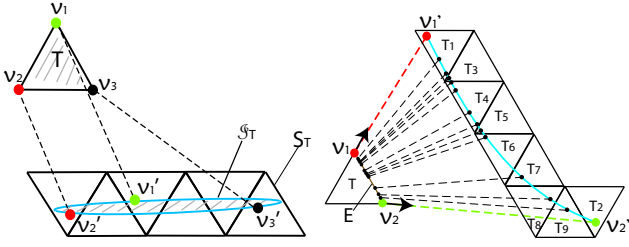


Fig. 10. This figure illustrates the stretching of a triangle  $T$  (left) and the notion of adaptive sampling on an edge  $E(v_1v_2)$  (right).  $T$  is the original triangle. The image of edge  $v_1v_2$  is  $v_1'v_2'$ . The dash lines show the mapping of the samples to the points on the image.

to the divergence of the reversed flow. Therefore, we can obtain at most three edges starting from  $T$  and will miss other edges. This will result in discontinuous Morse sets. The reason is we can not compute any sample inside the advected triangle  $\mathcal{S}_T$  (the tube in Figure 10, left) using vertex-based advection in this case. But we also notice that in these very distorted cases, the advected triangle  $\mathcal{S}_T$  will either intersect with the inner edges of the triangle strip  $S_T$  or situate inside a particular triangle (Figure 9, the first row). For the latter case, the forward map will compute the directed edges completely. For the former case, if we can sample the intersections of  $\mathcal{S}_T$  and the inner edges of  $S_T$ , then, we can compute the missing edges by calculating the backward map from the intersections. We thus introduce the algorithm of adaptive sampling along an edge to calculate such intersections.

---

#### Algorithm 1: Adaptive sampling on an edge

---

**Routine:** `adpative_edge_sampling(v1, v2, T1, original_T, neighbor_T, V, X, dτ, L)`

**Input:** two vertices  $v_1$  and  $v_2$ , triangle  $T_1$ , `original_T` and `neighbor_T`, vector field  $V$ , surface  $X$ , recursion level  $L$ , a user specified integral distance  $d_\tau$

**Output:** edges in  $DG_\tau$

**Local variables:** triangle  $t_2$ , stack  $s$

**Begin**

$L \leftarrow L + 1;$

if ( $L >$  maximum recursion level ||  $\|v_1 - v_2\|_2 <$  minimum distance)

$v_1 \leftarrow v_2; T_1 \leftarrow T_2;$

`new_edge(original_T, T1); new_edge(neighbor_T, T1);`

return  $v_1, T_1;$

$T_1 \leftarrow \text{trace}(v_1, d_\tau);$

$T_2 \leftarrow \text{trace}(v_2, d_\tau);$

if ( $T_1 == T_2$  || `share_edge(T1, T2)`)

$v_1 \leftarrow v_2; T_1 \leftarrow T_2;$

`new_edge(original_T, T1); new_edge(neighbor_T, T1);`

return  $v_1, T_1;$

else

`push(s, v2);`

$v_2 \leftarrow (v_1 + v_2)/2;$

`call adpative_edge_sampling(v1, v2, T1, original_T, neighbor_T, V, X, dτ, L);`

if ( $s \neq \text{NULL}$ )

$v_2 \leftarrow s$

`call adpative_edge_sampling(v1, v2, T1, original_T, neighbor_T, V, X, dτ, L);`

**End**

---

The idea of adaptive sampling on edges is explained as follows. We first trace from the two vertices of an edge  $E(v_1, v_2)$  (Figure 10, right). If the two triangles  $T_1$  and  $T_2$  containing the two advected vertices are the same triangle or they share a common edge, then we need not process  $E$  further. If they are not, it means that we need to place more samples on  $E$ . We compute more samples until we get a connected triangle strip containing the image of  $E$ . The reasoning behind this method is based on the fact that the result of a continuous object under

a continuous map should be continuous. More specifically, the image of an edge ( $v_1v_2$  in Figure 10, right) under the  $\tau$  map, a continuous map, should be a continuous curve ( $v_1'v_2'$  in Figure 10, right) which is contained by a connected triangle strip. We use a binary search along the edge to compute the intersection. We compute more samples if the two triangles  $T_1, T_2$  containing the two advected vertices  $v_1', v_2'$  do not share a common edge and  $T_1 \neq T_2$ . Figure 10 (right) demonstrates the idea of this algorithm. After combining this algorithm with the previous approach, we can compute a more accurate multi-valued map for a given flow. We wish to point out that due to a discrete representation, there is no guarantee of finding a continuous map under a highly divergent flow with a large  $d_\tau$ , even though we sample densely along the edges. However, we have not experienced with this issue.

Figure 11 indicates the difference in the MCGs obtained using the geometry-based map [2] and the method based on a  $\tau$  map. Both methods produce combinatorial multivalued maps that are outer approximations and hence produce valid Morse decompositions. However, as is indicated in Figure 7, the  $\tau$ -map approach leads to a combinatorial multivalued map  $\mathcal{F}$  with smaller images and hence a finer Morse decomposition. An extremely important point that can easily be overlooked is the freedom of choice in the construction of  $\mathcal{F}$ . We have chosen an approach that is a compromise between accuracy of images and speed of computation. For problems in which computational time is not a concern one can expand on the adaptive sampling technique and the choice of  $\tau$  (or  $d_\tau$ ) to refine the images. Alternatively, if one knows that the original vector field contains significant errors, then since the multi-valued map need only be an outer approximation, these errors can be incorporated into the construction of the images of  $\mathcal{F}$ . Thus, even in the presence of considerable noise one can guarantee that the resulting MCG is valid. Clearly, in this setting this need not be the case for any particular trajectory such as a periodic orbit or even a fixed point.

## 5 CONSTRUCTING AND VISUALIZING THE MCG

After obtaining the Morse decomposition  $M(X, V)$  of a vector field  $V$ , we compute the MCG of  $M(X, V)$  (Figure 4). Our implementation of Kalies and Ban's algorithm for MCG construction [10] is a two step process. In the first step, we represent the isolating neighborhood of each Morse set in  $M(X, V)$  as a node  $n_M$  in  $DG_\tau$  (or  $DG_g$  for geometry-based mapping). Then, we perform a breadth first search (BFS) starting from each of the compact node  $n_{M_i}$ . For each reachable node  $n_{M_j}$ , where  $i \neq j$ , we add an edge from  $n_{M_i}$  to  $n_{M_j}$  in MCG. In the second step, we minimize the MCG by removing all the redundant edges. For instance, if there are edges  $e_1 = (n_1, n_2)$ ,  $e_2 = (n_2, n_3)$  and  $e_3 = (n_1, n_3)$ , then the edge  $e_3$  is a *redundant edge*. We remove it from the MCG.

To visualize MCG, we need to classify the nodes. *Source Morse sets*,  $R_i$ , are nodes absent of incoming edges in the MCG. *Sink Morse sets*,  $A_i$ , are nodes with no outgoing edges in the MCG. *Saddle Morse sets*,  $S_i$ , are neither source Morse sets nor sink Morse sets. The  $R_i$ 's are colored green, the  $A_i$ 's are colored red and the  $S_i$ 's are colored blue. According to the partial order determined by the edges in MCG, we lay out the nodes such that the source Morse sets appear at the top of graph, the sink Morse sets are placed at the bottom of the graph and the saddle Morse sets are placed between the source and sink Morse sets. Figure 11 (a) and (b) display the MCGs of an analytical vector field. As a result, topological analysis and visualization can benefit from both graph theory and traditional geometric approaches.

Observe that some graphics applications may persue the individual trajectory-based vector field topology without being concerned with the fact that the obtained ECGs may not be reliable, for instance, the applications in texture synthesis [26, 27] and fluid simulation [21]. As an additional step of our pipeline of vector field analysis using Morse decomposition, an ECG [2] can be computed based on the obtained MCG (Figure 11 (c) and Figure 12 (c)).

## 6 RESULTS

In this section, we provide a number of vector field analysis results using the efficient Morse decomposition framework we introduced previously for both planar flow and surface-based flow. The vector fields

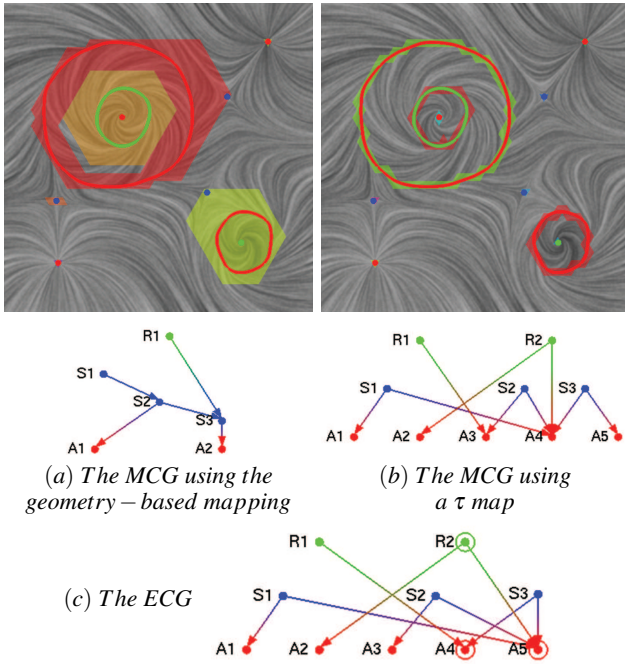


Fig. 11. This figure shows the Morse decompositions of an analytical data set. (a) The Morse decomposition of the vector field using the geometry-based map. (b) The Morse decomposition of the vector field using a multi-valued map with  $d_\tau = 0.8$ . The corresponding MCGs of the Morse decompositions are shown under the flow images. The greed dots indicate the source Morse sets, red dots refer to sink Morse sets and blue dots represent saddle Morse sets. (c) is the ECG of the vector field. In the ECG, the greed dots indicate the source or repelling periodic orbits, red dots refer to sink or attracting periodic orbits and blue dots represent saddles. We can see how the Morse sets are refined by using the idea of  $\tau$  map.

we show here are an experimental vector field created using the vector field design system introduced in [2], and two engine simulation data sets [14].

### 6.1 An Experimental Field

Figure 11 shows the Morse decomposition of a designed vector field. Ten Morse sets have been extracted using  $d_\tau = 0.8$ . The extraction takes 10.37 seconds on a 3.6 GHz PC with 3.0 GB RAM. The triangular mesh consists of 6144 triangles. Figure 11 (a) shows the MCG of the field using the geometry-based mapping, while (b) shows the MCG obtained using a  $\tau$  map. Both of these two MCGs can be used to compute the ECG of the flow (Figure 11, (c)). Compared to the three-layer structure of the ECG, an MCG is computational stable and has a multiple layer structure, which provides more information than the ECG. For instance, the MCG allows saddle to saddle connections, but an ECG does not. From the result, we observe that the geometry-based mapping approach is easy to compute, but tends to result in coarser Morse sets, while the MCG derived from a  $\tau$  map has finer Morse sets. Furthermore, if computed correctly, an ECG indicates the finest MCG, when the vector field has a finite number of fixed points and periodic orbits, all of which have an isolating neighborhood of their own [2].

### 6.2 The Engine Simulation Data Sets

The datasets we experiment on are the extrapolated boundary velocity fields and are obtained through simulation of in-cylinder flow. What engineers expect to observe is whether the flows on the surface follow the ideal patterns or not [14]. To investigate vector field analysis and visualization techniques can provide the critical insight that would have been difficult to obtain otherwise.

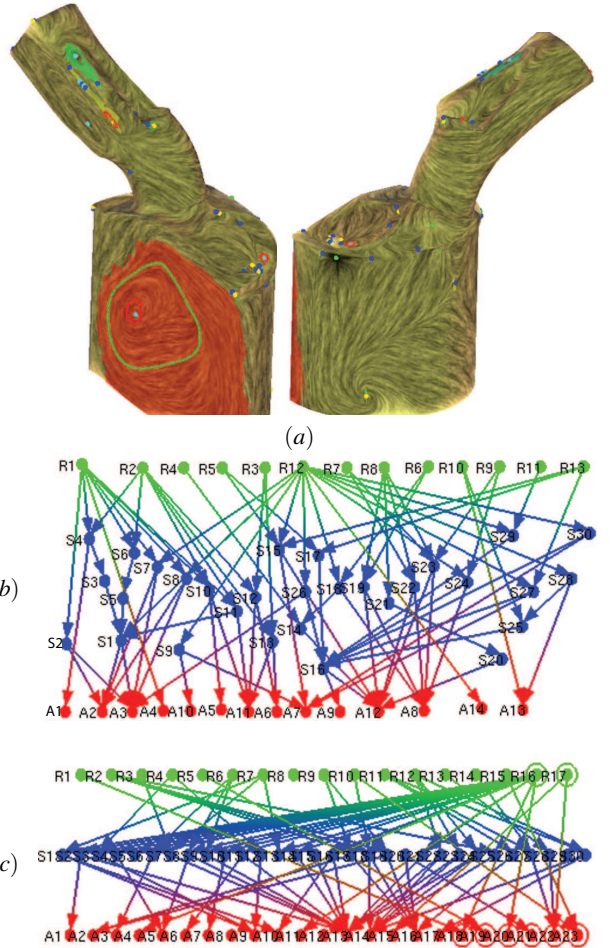


Fig. 12. This figure shows two viewpoints of the Morse decomposition of gas engine simulation data using a multi-valued map with  $d_\tau = 1$  (a). Note that the red region bounds the area of recirculating flow corresponding to tumble motion—an ideal pattern of motion with good mixing properties. (b) shows the MCG of the data. (c) displays its ECG.

The results are shown in Figures 3 (bottom), 12 and 13, respectively. In Figure 12, we observe a Morse set that has been extracted at the back of the in-cylinder of the chamber. It shows a recurrent pattern there which indicates the flow starting to approximate the ideal tumble motion. The Morse sets obtained based on a  $\tau$  map capture more desired regions with important features, while the approach using the geometry-based map could give rise of large Morse sets which will be difficult for engineers to identify the crucial features of the flow with certainty.

Figure 12 (a) shows two viewpoints of the Morse decomposition result from the gas engine simulation data with  $d_\tau = 1$ . (b) shows the MCG of the data, while (c) shows the ECG derived from the MCG. The gas engine surface geometry consists of 26298 triangles. There are a total of 57 Morse sets. The total time to perform Morse decomposition of this data is 220.48 seconds on a 3.6 GHz PC with 3.0 GB RAM. The number of the edges in  $DG_\tau$  is 244497. The result shown in Figure 13 is from the diesel engine simulation with  $d_\tau = 1$ . The number of triangles of the diesel engine surface geometry is 221574. There are 162 Morse sets being extracted. The time for the Morse decomposition computation for this data set is 252.67 seconds. The number of the edges in the  $DG_\tau$  is 1968629.



Fig. 13. A comparison of the obtained Morse sets of the diesel engine simulation data set. The first image shows the Morse sets obtained using the geometry based mapping. The two images to the right show the result using a multi-valued map with  $d_\epsilon = 1$  from two different view points. Note how much more refined the topological regions become.

## 7 CONCLUSION

In this paper, we have described an efficient computational framework for computing Morse decompositions of vector fields. Our approach makes use of a multi-valued map, an outer approximation of a  $\tau$  map, which encodes the dynamics of the vector field. Compared to individual trajectory-based vector field analysis, Morse decomposition and the associated MCG accounts for the numerical errors inherent in the vector field data, which makes it more suitable for a rigorous interpretation of vector field topology. In order to compute the multi-valued map efficiently, we make use of the idea of backward mapping as the complement of forward mapping, and introduce an adaptive sampling along the edges to account for the discontinuity problem while computing the approximate image. We show the utility of our approach in a number of applications including an analytical data in 2D plane and two engine simulation data sets on surfaces.

There are a number of future directions. First, the MCGs can be used to guide the pair wise cancellation based on Conley theory [2, 32]. Second, through the adaptive framework of computing multi-valued map proposed in this paper, a series of hierarchical MCGs can be computed. These hierarchical MCGs can be used to guide vector field clustering, vector field compression and automatic simplification. Finally, there is a need to extend the work to time-dependent vector fields for keeping track of the features of the flow and the applications of 3D vector fields as well.

## REFERENCES

- [1] E. Boczko, W. Kalies, and K. Mischaikow. Polygonal approximation of flows. 2006. (forthcoming), The online version available at: <http://www.math.fau.edu/kalies/papers/paf.pdf>.
- [2] 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*, 13, 2007. (forthcoming), Technical report available at: [http://eecs.oregonstate.edu/library/files/2006-27/vecfldedit\\_tr.pdf](http://eecs.oregonstate.edu/library/files/2006-27/vecfldedit_tr.pdf).
- [3] C. Conley. Isolated invariant sets and the morse index. *CMBS Regional Conference Series in Mathematics*, 38, 1978.
- [4] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press, Cambridge, MA, 1990.
- [5] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *Proceedings IEEE Visualization 94*, pages 140–147, 1994.
- [6] M. Eidenschink. *Exploring Global Dynamics: A Numerical Algorithm Based on the Conley Index Theory*. PhD thesis, Georgia Institute of Technology, 1996.
- [7] J. L. Helman and L. Hesselink. Representation and Display of Vector Field Topology in Fluid Flow Data Sets. *IEEE Computer*, 22(8):27–36, August 1989.
- [8] J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.
- [9] C. Johnson. Top Scientific Visualization Research Problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, July/August 2004.
- [10] W. Kalies and H. Ban. A computational approach to conley’s decomposition theorem. *Journal of Computational and Nonlinear Dynamics*, 1(4):312–319, 2006.
- [11] W. D. Kalies, K. Mischaikow, and R. C. A. M. VanderVorst. An algorithmic approach to chain recurrence. *Foundations of Computational Mathematics*, 5(4):409–449, 2005.
- [12] R. S. Laramee, C. Garth, H. Doleisch, J. Schneider, H. Hauser, and H. Hagen. Visual Analysis and Exploration of Fluid Flow in a Cooling Jacket. In *Proceedings IEEE Visualization 2005*, pages 623–630, 2005.
- [13] R. S. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology Based Flow Visualization: The State of the Art. In *The Topology-Based Methods in Visualization Workshop (TopoInVis 2005)*, 2006. proceedings to appear.
- [14] R. S. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques. In *Proceedings IEEE Visualization 2004*, pages 51–58, 2004.
- [15] K. Mischaikow. Topological techniques for efficient rigorous computation in dynamics. *Acta Numer.*, 11:435–477, 2002.
- [16] K. Polthier and E. Preuss. Identifying vector fields singularities using a discrete hodge decomposition. In *Mathematical Visualization III*, pages 112–134. Ed: H.C. Hege, K. Polthier, 2003.
- [17] 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*, 22(4):775–792, Dec. 2003.
- [18] G. Scheuermann, H. Hagen, H. Krüger, M. Menzel, and A. Rockwood. Visualization of Higher Order Singularities in Vector Fields. In *Proceedings IEEE Visualization '97*, pages 67–74, Oct. 1997.
- [19] G. Scheuermann, H. Krüger, M. Menzel, and A. P. Rockwood. Visualizing nonlinear vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, 1998.
- [20] G. Scheuermann, H. Krüger, M. Menzel, and A. P. Rockwood. Visualizing Nonlinear Vector Field Topology. *IEEE Transactions on Visualization and Computer Graphics*, 4(2):109–116, Apr./June 1998.
- [21] J. Stam. Flows on surfaces of arbitrary topology. In *ACM Transactions on Graphics (SIGGRAPH 03)*, volume 22, pages 724–731, July 2003.
- [22] H. Theisel, T. Weinkauff, H. Hege, and H. Seidel. Grid independent detection of closed stream lines in 2d vector fields. In *Proc. Vision, Modeling and Visualization 04*, 2004.
- [23] 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)*, pages 421–428, Nov. 2004.
- [24] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. In *ACM Transactions on Graphics (SIGGRAPH 03)*, volume 22, pages 445–452, July 2003.
- [25] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings IEEE Visualization 01*, pages 159–166, 2001.
- [26] G. Turk. Texture synthesis on surfaces. In *Proceedings of ACM SIGGRAPH 01*, pages 347–354, 2001.
- [27] L.-Y. Wei and M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of ACM SIGGRAPH 01*, pages 355–360, 2001.
- [28] T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(2), 2001.
- [29] T. Wischgoll and G. Scheuermann. Locating Closed Streamlines in 3D Vector Fields. In *Proceedings of the Joint Eurographics - IEEE TCVG Symposium on Visualization (VisSym 02)*, pages 227–280, May 2002.
- [30] T. Wischgoll, G. Scheuermann, and H. Hagen. Tracking Closed Streamlines in Time Dependent Planar Flows. In *Proceedings of the Vision Modeling and Visualization Conference 2001 (VMV 01)*, pages 447–454, Nov. 21–23 2001.
- [31] Y. Q. Ye, S. L. Cai, L. S. Chen, K. C. Huang, D. J. Luo, Z. E. Ma, E. N. Wang, M. S. Wang, and X. A. Yang. *Theory of limit cycles*, volume 66 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, second edition, 1986. Translated from the Chinese by C. Y. Lo.
- [32] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.