Efficient Morse Decompositions of Vector Fields

Guoning Chen, Konstantin Mischaikow, Robert S. Laramee, *Member, IEEE Computer Society,* and Eugene Zhang, *Member, IEEE Computer Society,*

***Abstract*— Existing topology-based vector field analysis techniques rely on the ability to extract the 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. We advocate the use of Morse decompositions, which are robust with respect to perturbations, to encode the topological structures of a vector field in the form of a directed graph, called a Morse connection graph (MCG).**

While an MCG exists for every vector field, it need not be unique. Previous techniques for computing MCGs, while fast, are overly conservative and usually result in MCGs that are too coarse to be useful for the applications. To address this issue, we present a new technique for performing Morse decomposition based on the concept of τ-*maps*, which typically provides finer MCGs than existing techniques. Furthermore, the choice of τ provides a natural tradeoff between the fineness of the MCGs and the computational costs.

We provide efficient implementations of Morse decomposition based on τ-maps, which include the use of forward and backward mapping techniques and an adaptive approach in constructing better approximations of the images of the triangles in the meshes used for simulation. Furthermore, we propose the use of *spatial* τ-*maps* in addition to the original *temporal* τ-*maps*. These techniques provide additional tradeoffs between the quality of the MCGs and the speed of computation. We demonstrate the utility of our technique with various examples in the plane and on surfaces including engine simulation datasets.

***Index Terms*— Vector field topology, uncertainty, Morse decomposition, τ-maps, Morse connection graph, flow combinatorialization.**

I. Introduction

Extracting and visualizing vector field topology has important applications in Computational Fluid Dynamics (CFD) [15], weather prediction, tsunami and hurricane modeling, and airplane design and control. For instance, the existence of recirculation zones (periodic orbits) can indicate stagnant flow which may be undesirable in engine design, because stagnant flows indicate trapped heat in the engine [16].

Past work defines the topology of two-dimensional vector fields as fixed points and periodic orbits as well as the separatrices that connect them [10], [4]. This leads to a graph representation of the vector field which is referred to as *Entity Connection Graph*, or *ECG*. However, analysis and visualization of vector field topology based on individual trajectories can raise questions with respect to interpretation as the discrete nature of fluid flow data poses several challenges. First, data samples are only given at discrete locations, such as cell vertices or cell centers. Interpolation schemes are then used to reconstruct the vector field between the given samples. Second, the given data samples themselves are numerical approximations, e.g., approximate solutions to a set of partial differential equations. Third, 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 the flow field? Could the error inherent to multiple numerical approximations produce misleading information? Figure 1 provides examples in which proper interpretation can be difficult when performing analysis based on individual trajectories.

Figure 1(a) shows an analytical vector field which contains pitchfork bifurcation [9]. The results shown in the two columns of (a) are obtained by computing sample vector values using two different meshes: (left) a regular triangulated mesh with 6144 triangles, and (right) a triangulated mesh with 1000 triangles. Notice that using different meshes leads to different ECGs (third row of Figure 1(a)). Figure 1(b) demonstrates a saddle-saddle connection bifurcation [9]. The images to the left of Figure 1(b) show the original flow, while the images to the right show the flow that was obtained from the original one after introducing a small amount of perturbation (we have randomly perturbed the vector direction at each vertex by an angle between 0° and 1°.)Notice that ECGs (third row of Figure 1(b)) are sensitive to noise. Figure 1(c) provides a case of Hopf-bifurcation [15]. The image to the left of Figure 1(c) (second row) shows the resulting topology using an adaptive fourth-order Runge-Kutta integration, while the image to the right illustrates the topology of the same vector field using a second-order Runge-Kutta integration [2] [23]. This clearly demonstrates that the ECGs rely on the employed numerical scheme. (The ECGs in all the example flows are computed using the algorithms proposed by Chen et al. [4].) These observations motivate the study of a more reliable way of defining and extracting vector field topology than the existing techniques. We point out that addressing such uncertainty in visualization was identified as one of the most important future challenges by Johnson [12].

In order to address this important challenge we present a rather different approach to the representation, extraction and visualization of flow topology. The representation of the global dynamics is done in terms of an acyclic directed graph called the *Morse connection graph* (MCG). The nodes in this graph, which we refer to as *Morse sets*, correspond to polygonal regions in the

G. Chen is with the School of Electrical Engineering and Computer Science, Oregon State University, 1148 Kelley Engineering Center, Corvallis, OR 97331,USA.

Email:chengu@eecs.oregonstate.edu.

K. Mischaikow is with the Department of Mathematics, Hill Center-Busch Campus, Rutgers, The State University of New Jersey 110 Frelinghusen Rd Piscataway, NJ 08854-8019, USA.

Email:mischaik@math.rutgers.edu.

R.S. Laramee is with the Department of Computer Science, Swansea University, SA2 8PP, Wales, United Kingdom.

Email:R.S.Laramee@swansea.ac.uk.

E. Zhang is with the School of Electrical Engineering and Computer Science, Oregon State University, 2111 Kelley Engineering Center, Corvallis, OR 97331, USA.

Email:zhange@eecs.oregonstate.edu.

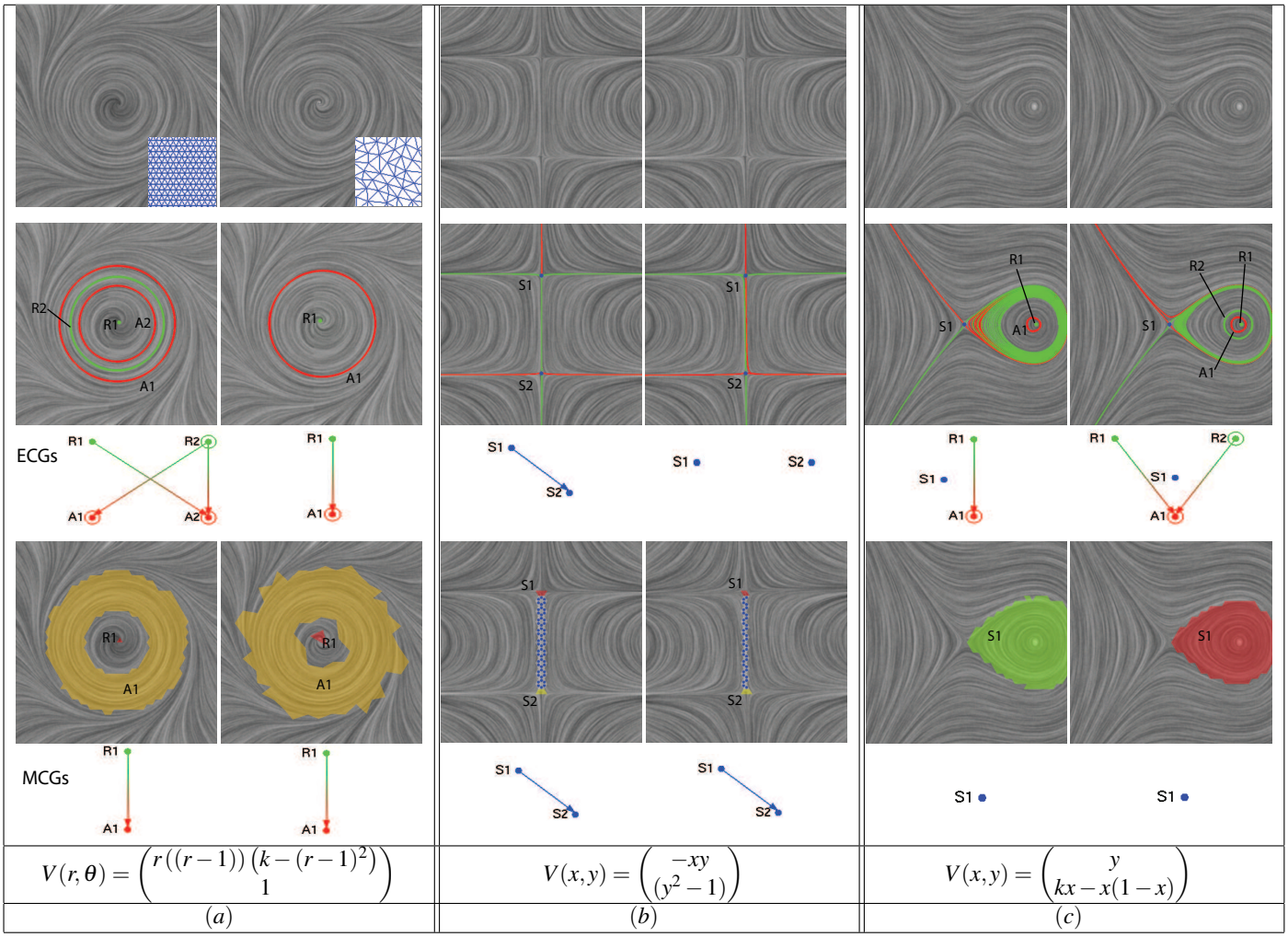

Fig. 1. Examples of the instability of individual trajectory-based vector field topology analysis (i.e. ECGs) due to the choice of discretization scheme (a), noise (b), and the error from numerical integration scheme (c). (a) shows a vector field containing pitchfork bifurcation ($k = 0.05$). It illustrates the deviated ECGs obtained under two different discretization schemes. The vector field shown in (b) is an example flow having saddle-saddle connection. The two ECGs are computed based on the original result flow and its perturbation version. It illustrates the possible influence of the unexpected noise in the data. (c) uses a vector field with Hopf-bifurcation ($k = 0.0025$) to illustrate that ECGs can be different using different numerical schemes. The image to the left shows the resulting topology using the adaptive fourth-order Runge-Kutta integration, while the image to the right shows the topology of the same flow using a second-order Runge-Kutta integration. In the ECGs of all the example fields, green dots indicate the source or repelling periodic orbits, red dots refer to sink or attracting periodic orbits and blue dots represent saddles. The two bottom rows provide the results of Morse decompositions and the associated MCGs of these fields using the idea of τ-maps proposed in this paper. The τ's for these fields are 40, 20 and 80, respectively. Note that for all the examples shown here, the MCGs are stable. The colored regions in the flow-like images (the fourth row) are the Morse neighborhoods of the extracted Morse sets. Different colors indicate different Morse sets. The color-dotted regions indicate the connection between Morse neighborhoods. Constrained to the underlying mesh, the Morse neighborhoods and the regions showing the connections between Morse neighborhoods may look irregular (e.g. the Morse neighborhoods in (a) right and (c) left, and the connection regions in (b)). In the MCGs, green dots stand for the source Morse sets, red dots for the sink Morse sets and blue dots for the saddle Morse sets.

phase space, which we define to be *Morse neighborhoods*. All the recurrent dynamics is contained in the Morse neighborhoods. The edges in an MCG indicate how the flow moves from one Morse neighborhood to another. In contrast to trajectory-based topological analysis, such as vector field skeleton and ECG, an MCG is stable with respect to perturbations, i.e. given sufficient information on errors of the vector field it is possible to make rigorous interpretations about the underlying dynamics [14]. In other words, a well defined error, $\varepsilon > 0$, can be bounded and included into the map of the flow domain. We demonstrate the stability of MCGs in Figure 1 (the last two rows).

To perform Morse decomposition, i.e., compute MCGs, Chen et al. [4] first construct another directed graph by considering the behaviors of the vector field along edges of the triangles, which we refer to as the *geometry-based method*. We refer to the process of encoding the flow dynamics into a directed graph as *flow combinatorialization*. Because the triangulation is not adapted to the vector field, this can result in coarse Morse sets (Figure 2(b)). In this paper we exploit a temporal discretization, which we refer to as a *τ-map*, that is obtained by integrating a finite set of points for a finite amount of time. Theoretically this method can produce as detailed an MCG as is desired and in practice it produces a finer MCG (Figure 2(c) (d)) than the geometry-based method. The key challenges with the τ-map guided approach are choosing an appropriate temporal discretization of the flow and constructing a high-quality flow combinatorialization, which is the discrete outer approximation of a τ-map. In our implementations, we will compute it as a directed graph, denoted by $\mathscr{F}_\tau$ under a time τ.

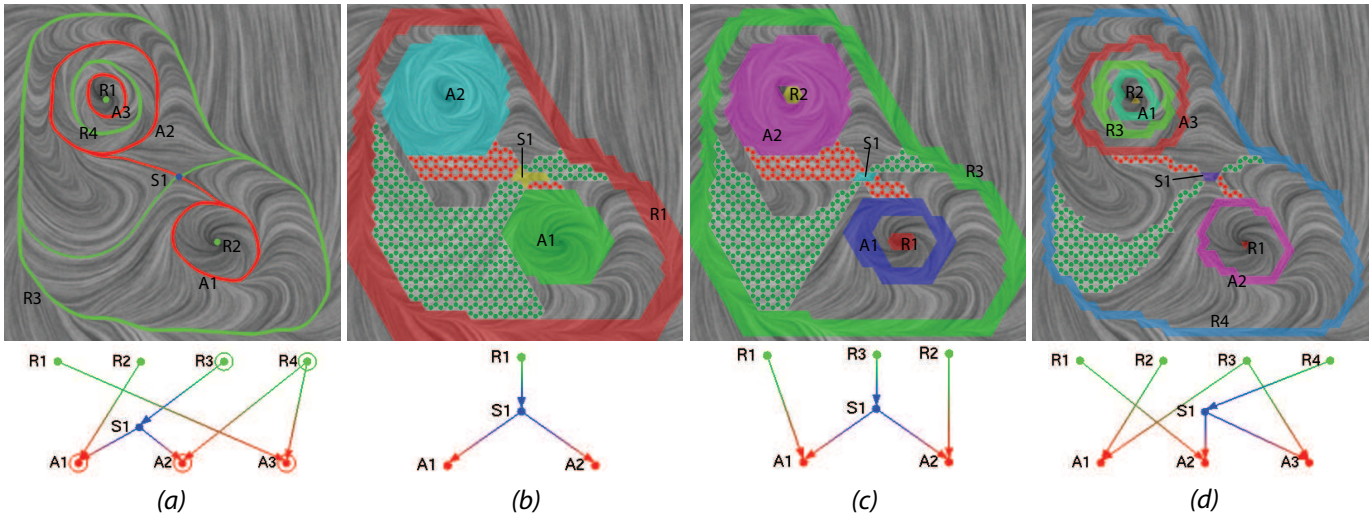

Fig. 2. This figure shows the various analysis results of an experimental field using ECG and MCGs, respectively: (a) ECG, (b) MCG (geometry-based method), (c) MCG ($\tau = 6$), (d) MCG ($\tau = 24$). The computation time for (b-d) is 0.14s, 1.78s, and 4.31s, respectively. Observe that the larger the τ used, the better (closer to optimal) the Morse decompositions, but the time for computing the Morse decomposition increases accordingly. The coloring scheme of the MCG is described in Figure 1. Notice that the graphs shown in (a) and (d) are essentially the same although they are labeled differently. The execution time was measured on a 3.0 GHz PC with 1.0 GB RAM. The color-dotted regions indicate the connections between a saddle Morse set to another Morse set: source (green), sink (red), and saddle (blue).

From it we extract the Morse sets.

The work presented here yields the following benefits and contributions:

1) We present a theoretically sound framework based on Morse decompositions from which more rigorous statements can be made with respect to the extraction of flow topology than the individual trajectory-based analysis (Figure 1).
2) We provide a means to obtain finer Morse decompositions of a given vector field than the geometry-based method by using the idea of τ-maps. A directed graph $\mathscr{F}_\tau$ obtained using flow combinatorialization based on a τ-map, is introduced, which we use to perform Morse decomposition and compute the MCG. We also explore the effect of the values of τ on the level of details of the Morse decompositions (Figure 2).
3) We introduce a computationally tractable implementation of the efficient construction of the $\mathscr{F}_\tau$ and consequently fine Morse decompositions (Section IV-B).
4) We explore the use of spatial τ_s vs. temporal τ, i.e., tracing particle for a finite distance instead of time, which provides domain experts an alternative to the temporal τ-maps to analyze their data (Section V).
5) We apply the proposed topological analysis technique to both analytical data and application-oriented data sets including engine simulation data from CFD on 3D surfaces (Section VI).

The rest of this paper is organized as follows. Section II provides a brief review of related work on vector field topology analysis. Section III introduces the methodology of vector field analysis using Morse decompositions and the concept of τ-maps. The pipeline of Morse decomposition is also presented in Section III. Section IV describes a number of practical algorithms to perform flow combinatorialization. Section V proposes the use of spatial τ_s in order to achieve faster construction of flow combinatorialization. Section VI shows the utility of our approach to the engine simulation data followed by a summary and discussion of future work in Section VII.

II. Related Work

Helman and Hesselink introduced to the visualization community the notion of flow topology [10] [11]. Since then, much research has been done on topological analysis of vector fields in the past two decades. To review all of it is beyond the scope of this paper. Here, we briefly review the most closely related work to this paper. We refer interested readers to a number of surveys [15], [22].

A. Vector Field Topology

Much work has been done to address the extraction of vector field topology in two-dimensional vector fields. Tricoche et al. [30] and Polthier and Preuß [21] give efficient methods to locate fixed points in a vector field. Scheuermann et al. use *Clifford algebra* to study the non-linear fixed points of a vector field [25] and present the approaches of visualizing higher-order fixed points and non-linear topology of a given vector field [24], [25]. Wischgoll and Scheuermann [34] present an algorithm for detecting periodic orbits in planar flows. They also extend this work to 3D vector fields [35] and time-dependent flows [36]. Theisel et al. [28] present a mesh independent approach to compute periodic orbits.

In general, previous topology-based techniques are based on individual trajectories. As we have seen in Figure 1, this may lead to ambiguous interpretation of the given data. To address this issue, we advocate the use of Morse decomposition. Compared to the individual trajectory-based methods, Morse decomposition takes the errors introduced during simulation and analysis into consideration given a certain error bound, and leads to a more rigorous interpretation of the given data.

B. Morse Decomposition and Conley Theory

Conley index theory and Morse decomposition have been introduced to the scientific visualization community by Zhang et al. [38] and Chen et al. [4], respectively. Morse decomposition focuses on extraction and analysis of the invariant sets of a

flow. Fixed points and periodic orbits are examples of invariant sets. An index called the *Conley index* [4] [19], identifies the type of the invariant set. Efficient algorithms also exist for identifying the neighborhoods of the invariant sets [14]. Chen et al. [4] implement a Morse decomposition algorithm using the geometry-based method. This approach guarantees to produce Morse decompositions but it is typically coarser than what is required (Figure 2(b)). We address this by employing the idea of τ-maps (Section III-D). Compared to the geometry-based method, the τ-map guided method introduced in this paper gives rise to a directed graph ($\mathscr{F}_\tau$) which encodes the dynamics of the given data more accurately [14]. The MCGs extracted from $\mathscr{F}_\tau$ are finer than those from the geometry-based method (Figure 2(d)).

III. Background

In this section, we present a compact summary of the theories of dynamical systems upon which our work is built. Our discussion will focus on time-independent flow only in this paper.

A. Entity Connection Graphs (ECGs)

A vector field defined on a manifold M can be expressed in terms of a differential equation $\dot{x} = f(x)$. The set of solutions to it gives rise to a *flow* on M; that is a continuous function $\varphi : \mathbf{R} \times M \to M$ satisfying $\varphi(0,x) = x$, for all $x \in M$, and

$$\varphi(t, \varphi(s,x)) = \varphi(t+s,x) \tag{1}$$

for all $x \in M$ and $t,s \in \mathbf{R}$. Given $x \in M$, its *trajectory* is

$$\varphi(\mathbf{R},x) := \cup_{t\in\mathbf{R}} \varphi(t,x). \tag{2}$$

$S \subset M$ is an *invariant set* if $\varphi(t,S) = S$ for all $t \in \mathbf{R}$. Observe that for every $x \in M$, its trajectory is an invariant set. A point $x \in M$ is a *fixed point* if $\varphi(t,x) = x$ for all $t \in \mathbf{R}$. More generally, 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*.

Because we are considering systems with invariant sets such as periodic orbits, the definition of the limit of a solution with respect to time is non-trivial. The *alpha* and *omega limit sets* of $x \in M$ are

$$\alpha(x) := \cap_{t<0} \mathrm{cl}(\varphi((-\infty,t),x)), \quad \omega(x) := \cap_{t>0} \mathrm{cl}(\varphi((t,\infty),x))$$

respectively (cl stands for the closure).

Given a point $x_0 \in M$, its trajectory is a *separatrix* if the pair of limit sets $(\alpha(x_0), \omega(x_0))$ consists of a saddle fixed point and another object that can be a source, a sink, or a periodic orbit.

The graph illustrating the connectivity between fixed points and periodic orbits is called the *entity connection graph* (ECG) [4] (Figure 1 (second row), Figure 2(a)). As Figure 1 indicates, ECGs are sensitive to noise as well as the choices of interpolation scheme, numerical integration technique, and sampling patterns.

B. Morse Connection Graphs (MCGs)

We are interested in describing the topological structures of the flow generated by a vector field $\dot{x} = f(x)$ defined on a triangulated surface $X \subset M$. However, the information we are given consists of a finite set of vectors

$$\{f_d(v_i) \mid v_i \text{ a vertex of } X\} \tag{3}$$

obtained either by a numerical simulation or from experiment. 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,

$$||f(v_i) - f_d(v_i)|| \leq \varepsilon. \tag{4}$$

In addition, since we are only given the data (Eq. 3) 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. 4) 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 Λ. We assume that for each $\lambda \in \Lambda$, the vector field $\dot{x} = F(x,\lambda)$ gives rise to a flow $\varphi_\lambda : \mathbb{R} \times X \to X$.

In this setting we assume that there exist 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 φ_{λ_0} and φ_{λ_1} need not agree [9]. 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. This leads us to weaken the topological structures which we use to classify the dynamics.

A *Morse decomposition* of X for a flow φ_λ is a finite collection of disjoint compact invariant sets, called *Morse sets* [14]

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

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

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

It is easy to verify that any structures associated with recurrent dynamics of φ_λ, i.e. fixed points, periodic orbits, chaotic dynamics, must lie in the Morse sets [14]. 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 $\mathscr{P}_\lambda$ 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 connection graph and denoted by MCG_λ [4] (Figures 1, 2 bottom rows). Moreover, without worrying about the potential noise and numerical errors, 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 [4]. Though, there may not exist a finest Morse decomposition. Consider the flow generated by the differential equation $x' = x^2 \sin(1/x)$. It has an infinite number of isolated fixed points and hence there is no finest Morse decomposition (remember that there can only be a finite number of Morse sets). Any Morse decomposition of it can be refined further.

C. MCG Construction

We now summarize the pipeline of constructing an MCG given the vector field V defined on a triangulated surface X.

First, we perform flow combinatorialization. That is, we encode the flow dynamics into a directed graph, denoted by $\mathscr{F}$, whose nodes represent the elements (e.g. triangles) of the underlying

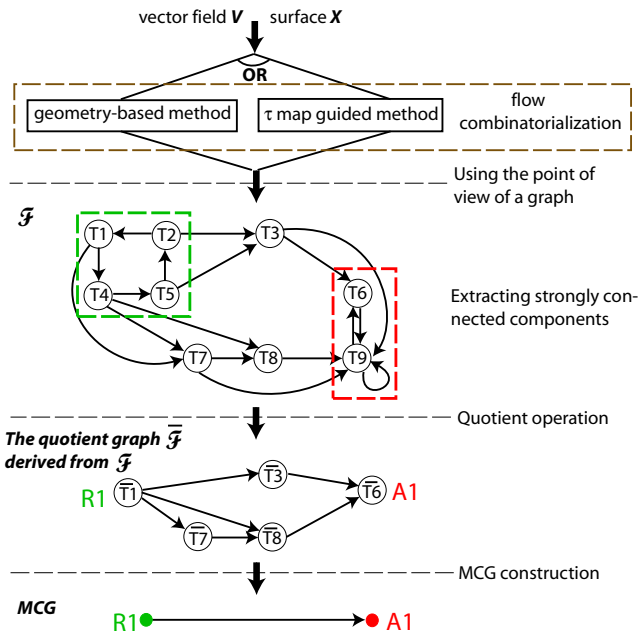

Fig. 3. This figure illustrates the pipeline of MCG construction. We first compute $\mathscr{F}$ (top) based on the underlying flow. The edges in the multi-valued map demonstrate the mapping relations of the polygons. Based on the $\mathscr{F}$, we extract the strongly connected components, which represent either the Morse sets (middle-top, inside colored boxes) or intermediate nodes that describe gradient-like behaviors (middle-top, T3, T7, T8). We then collapse each strongly connected component of the $\mathscr{F}$ into a single node to obtain a quotient graph $\overline{\mathscr{F}}$. Note that the nodes in this graph correspond to either Morse sets or the polygonal regions of gradient-like flow behaviors (i.e. trivial Conley index). Finally, the MCG (the bottom graph) is obtained by collapsing nodes with trivial Conley index and removing redundant edges.

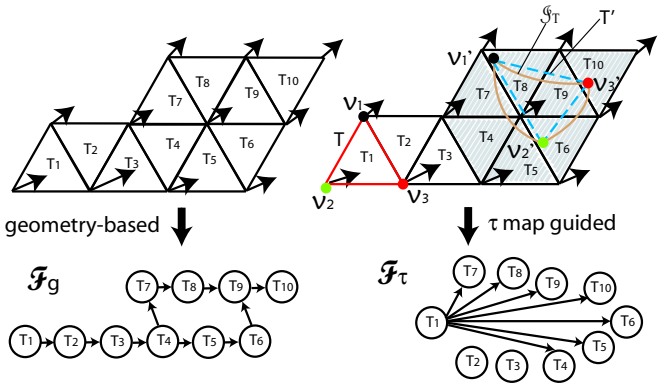

Fig. 4. This figure compares two ways of performing flow combinatorialization: (left) geometry-based method, and (right) τ-maps. In the directed graphs, each node corresponds to a triangle of the mesh. The red triangle $T = T_1$ is the starting triangle, the light brown curved closure is the real image of T, the blue dashed triangle is the approximation of the real image.

mesh and edges indicate the flow dynamics, i.e., an edge from triangle T_i to triangle T_j indicates that $\varphi(T_i) \bigcap T_j \neq \emptyset$ (Figure 4, left). The details of this will be described in Section III-D.

Second, we find the strongly connected components of the directed graph $\mathscr{F}$, which gives rise to the Morse neighborhoods that are the polygonal regions constrained by the given mesh in the phase space. They contain the Morse sets $M(X,V)$ of the flow and have a non-trivial Conley index [14] (Figure 3, middle-top).

Third, we compute a quotient graph $\overline{\mathscr{F}}$ from $\mathscr{F}$ by treating each strongly connected component of $\mathscr{F}$ as a node (Figure 3, middle-bottom). The nodes in this quotient graph $\overline{\mathscr{F}}$ include Morse sets (non-trivial Conley index) and the intermediate nodes corresponding to the polygonal regions with gradient-like flow behaviors (i.e. trivial Conley index). An edge $\overrightarrow{\overline{m}\overline{n}}$ in $\overline{\mathscr{F}}$ indicates that there is at least one edge $\overrightarrow{kl}$ in $\mathscr{F}$ such as $\overline{k} = \overline{m}$ and $\overline{l} = \overline{n}$.

Finally, we extract the MCG from $\overline{\mathscr{F}}$ by removing intermediate nodes from $\overline{\mathscr{F}}$ as illustrated in Figure 3 (the bottom graph). The algorithm for MCG construction can be found in [13].

To visualize the MCG, we classify the nodes of the MCG into three types: *Source Morse sets*, R_i, are nodes absent of incoming edges in the MCG; *Sink Morse sets*, A_i, are nodes without 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 the MCG, we lay out the nodes such that the source Morse sets appear at the top of the 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. Figures 1, 2 and 12 display the MCGs of a number of analytical vector fields. Compared to the three-layer structure of the ECG, an MCG has a multi-layer structure, which provides more information than the ECG. Furthermore, unlike ECGs, saddle-saddle connection is a generic case in MCG (Figure 1(b), Figure 12(b)). Note that finer classification of Morse sets, e.g., Saddle Morse sets, can be realized based on Conley index theory [19].

We wish to emphasize that some graphics applications may pursue the individual trajectory-based vector field topology without being concerned with the fact that the obtained ECGs may not be topologically rigorous, such as, the applications in texture synthesis [31], [33] and fluid simulation [27]. For such applications, an ECG can still be extracted from Morse decomposition as an additional step [4].

D. Flow Combinatorialization Based on τ-maps

We now turn to the issue of flow combinatorialization, i.e., the process of generating the graph $\mathscr{F}$ based on a vector field V defined on a triangulated mesh X. Chen et al. [4] present a geometry-based approach as follows: The vertices of the directed graph $\mathscr{F}$ correspond to the triangles of the mesh. The edges of $\mathscr{F}$ are obtained by considering the flow behavior across each edge of each triangle. An edge $T_i \rightarrow T_j$ in $\mathscr{F}$ indicates the flow can enter from T_i to T_j, where T_i and T_j are neighboring triangles (Figure 4, left). We refer to the resulting directed graph as $\mathscr{F}_g$. An MCG can then be obtained from $\mathscr{F}_g$ using the pipeline described in Section III-C.

Since the mesh is not fitted to the flow, this approach is not guaranteed to obtain the correct dynamics of the flow. In our experiments, we have found that it often results in a rather coarse outer approximation of the underlying dynamics, i.e., Morse sets that contain multiple fixed points and periodic orbits (Figure 2(b)) or no structures at all (Figure 15 left column). This makes subsequent analysis and physical interpretation less effective. To obtain the Morse decomposition that are closer to optimal, we introduce the concept of τ-maps, which allows us 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* τ-time discretization *of the flow* φ *is a map* $f_\tau : X \rightarrow X$ *defined*

by

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

We refer to this map as a τ-map. Thus, finding Morse decompositions for the flow φ is equivalent to finding Morse decompositions for f_τ.

The fact that X is a triangulated surface provides us with an appropriate discretization in space. Let $\mathscr{X}$ be the triangulation of X (i.e., a set of triangles). We will approximate f_τ using a *combinatorial multi-valued map* $\mathscr{F} : \mathscr{X} \rightrightarrows \mathscr{X}$, that is a map such that for each triangle $T \in \mathscr{X}$, its image is a set of triangles, i.e. $\mathscr{F}(T) \subset \mathscr{X}$.

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

$$f_\tau(T) \subset \mathrm{int}\,(|\mathscr{F}(T)|)$$

for every $T \in \mathscr{X}$ where $|\mathscr{F}(T)| := \cup_{R \in \mathscr{F}(T)} R$, *int* denotes the interior. As an example, we refer readers to Figure 4 (right). In this example, we assume that the true image of the triangle $T = T_1$ is $\mathscr{I}_T$. It is obtained by advecting T according to the underlying flow over a time τ. According to the definition, the outer approximation of $\mathscr{I}_T$ is the set of triangles $T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$. Mathematically, we say that T has been mapped to *multiple* triangles of the same mesh by a function (or a map) f_τ that is determined by the underlying flow under a certain time τ.

From the point of view of computation it is useful to view $\mathscr{F}$ as a directed graph, which we denote as $\mathscr{F}_\tau$. (Figure 4, right). Similar to $\mathscr{F}_g$, the vertices, T_i, of an $\mathscr{F}_\tau$ are the triangles of the underlying mesh and the edges indicate the outer approximation of the images of the triangles over time τ. For instance, an edge $T_i \to T_j$ indicates that the image of the triangle T_i over time τ will intersect with the triangle T_j (Figure 4, right).

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

E. The Stability of MCGs

The definition of an outer approximation and the fact that the triangles in the strongly connected components of $\mathscr{F}$ form isolating neighborhoods for the Morse sets demonstrate why the MCG remains constant under small perturbations of the vector field (Figure 1(b)). Since f_τ is a continuous map and each triangle T is compact, the image $f_\tau(T)$ is a compact set. If $\mathscr{F}$ is an outer approximation, then by definition $f_\tau(T)$ is contained in the interior of the set $|\mathscr{F}(T)|$. Thus, this property will also hold for any sufficiently small perturbation of f_τ, which means that given a multi-valued map for f_τ we have the same $\mathscr{F}_\tau$ for any sufficiently small perturbation of f_τ. Figure 5 provides an illustrative example to explain this property of an outer approximation. In this figure, A triangle $T = T_1$ is advected according to the original flow (represented by the black arrows). Its image $\mathscr{I}_T$ is shown as the closure bounded by a blue curve. It intersects with a set of triangles (the shaded triangles) $T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$ of the mesh, which forms the outer approximation of $\mathscr{I}_T$. When we artificially introduce a random perturbation for each vector value (shown as magenta arrows) and advect the triangle T under the new flow, we obtain a new image $\mathscr{I}'_T$ of it (shown as magenta dashed curved closure). If we bound the perturbation of each vector to guarantee that the new obtained image $\mathscr{I}'_T$ will intersect the same set of the triangles as the $\mathscr{I}_T$ obtained under the original vector field, we will obtain the same outer approximation of the image of T. Hence, the corresponding portion of the directed graph $\mathscr{F}_\tau$ will not change. The MCG is consequently stable. In other words, the outer approximation provides more space for error in the given data. We also point out that the MCGs obtained using the geometry-based method are also stable. Consider the example shown in Figure 5. Note that the flow behavior across each edge of the mesh does not change after a smaller perturbation, neither does the corresponding portion of $\mathscr{F}_g$. Therefore, the MCG remains the same. On the other hand, in this setting this need not be the case for any particular trajectory such as a periodic orbit or even a fixed point. That is, a particular trajectory may be changed after any perturbation. Of course, we can go one step further and insist that an ε-neighborhood of $f_\tau(T)$ be contained in $|\mathscr{F}(T)|$. We will in general get a coarser $\mathscr{F}_\tau$, but the resulting Morse decompositions will be valid for any vector field whose τ-map lies within ε of f_τ.

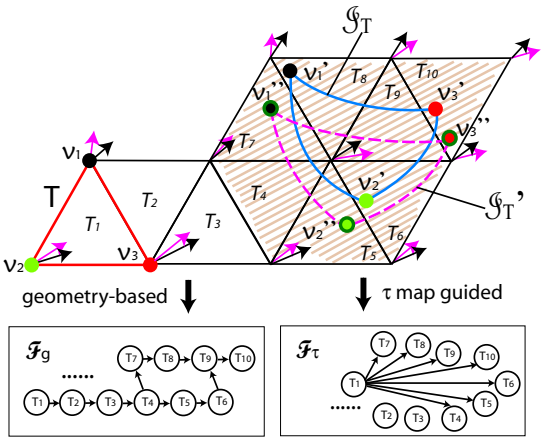

Fig. 5. This figure illustrates that using outer approximation, Morse decomposition is stable under certain error bound ε. The original image of triangle T is $\mathscr{I}_T$ (region inside the blue curve), and the outer approximation of that is the set of shaded triangles $T_4, T_5, T_6, T_7, T_8, T_9, T_{10}$. After a random perturbation (shown by magenta arrows) of the original field (shown by black arrows), we recalculate the image $\mathscr{I}'_T$ of triangle T, which is shown as the magenta curved closure. Although it is different from $\mathscr{I}_T$, the outer approximation consists of the same set of the triangles. Therefore, the corresponding portion of the direction graph $\mathscr{F}_\tau$ remains the same. Hence, we say that Morse decomposition is stable under an error bound ε, which here is the maximal allowed perturbation that will not change the outer approximation of the image of each triangle.

After applying the idea of τ-map based Morse decomposition to the analytical field shown in Figure 2, we obtain a finer Morse decomposition (Figure 2(d)). The colored regions there indicate the isolating neighborhoods of the Morse sets. Different color regions indicate different Morse sets. The flow-like texture regions without color indicate the regions of gradient-like flow (Section III-B). The color-dotted regions indicate the connections between a saddle Morse set to another Morse set: source (green), sink (red), and saddle (blue).

IV. Our Algorithm for Flow Combinatorialization

In this section, we describe a practical algorithm for performing flow combinatorialization based on τ-maps. First, we explain the computational model used in this paper. The underlying domain of

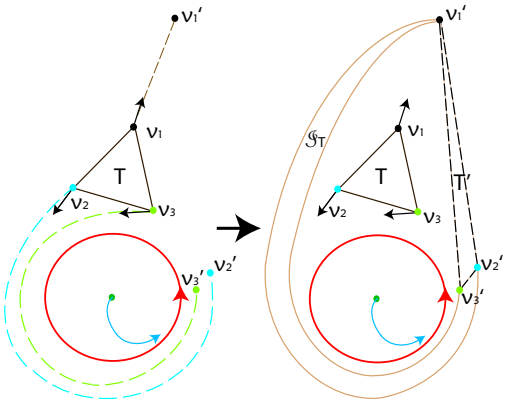

Fig. 6. This figure demonstrates a case of a distorted image of a triangle T consisting of v_1, v_2 and v_3 using a large τ. The red closed loop represents a periodic orbit. The colored dash lines in the left figure show the trajectories of the two vertices. The light brown curved closure in the right figure shows the real image $\mathscr{I}_T$ of the original triangle T, while the dash triangle T' is the approximate image.

this model is represented by a triangulated 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 piecewise linear interpolation. On curved surfaces, we adopt the interpolation scheme of Zhang et al. [38] which guarantees vector field continuity across the vertices and edges of the mesh. Vector field continuity is required for rigorous topological analysis.

As we have seen in Section III, when employing the idea of τ-maps, computing the correct flow combinatorialization $\mathscr{F}_\tau$ is the most crucial step in the pipeline of the Morse decomposition. To obtain an accurate $\mathscr{F}_\tau$, it is essential to compute the accurate (sufficient) outer approximation of the image of each triangle of the given mesh and obtain the directed edges of $\mathscr{F}_\tau$ accordingly. In this section, we introduce several methods to compute the outer approximation efficiently.

A. Explicit Outer Approximation Computation

1) A Rigorous Method: This method is applicable to any τ-time discretization and produces a rigorous outer approximation assuming that a bound ε on the errors in the underlying vector field is known. Given a triangle T one covers it with squares of size ε. For each square S define $\tau_*(S) = \min\{\tau(x) \mid x \in S\}$ and $\tau^*(S) = \max\{\tau(x) \mid x \in S\}$. Using rigorous enclosure techniques [1], [20] one obtains an outer enclosure $\mathscr{I}_S$ of the true image of the square S integrated forward for all times $\tau_*(S) \le t \le \tau^*(S)$. Then $\mathscr{I}_T = \cup \mathscr{I}_S$, where the union is taken over all squares S, is an outer approximation for $f_\tau(T)$.

This method is computationally costly. First, the number of squares needed to cover the triangle T is of order ε^{-2}, which for small ε is large. Second, due to the Gronwall inequality [9] the size of the image of $\mathscr{I}_S$ grows exponentially as a function of the integration time. Thus, to get tight outer approximations one must choose small ε. On the other hand, variants of this method have been used to obtain rigorous computer assisted proofs in dynamics [1].

2) An Accurate Outer Approximation Through Outer Boundary Tracking: Because we are working with flows, if we use a τ-time discretization which is a constant function, then f_τ is a diffeomorphism which is homotopic to the identity map. Consider a triangle T and its boundary ∂T. Then $f_\tau(\partial T)$ is the boundary of $f_\tau(T)$. Thus, it is sufficient to have a rigorous outer approximation of $f_\tau(\partial T)$ to obtain a rigorous outer approximation of $f_\tau(T)$.

To compute the outer approximation of $f_\tau(\partial T)$, we first consider using the three vertices of a triangle as the sample points. For instance, in Figure 4 we trace the three vertices v_1, v_2, v_3 of T with τ time and obtain v_1', v_2', v_3'. They form an imaginary triangle T', that intersects with a set of triangles T_i. They form an outer approximation of the true image, $\mathscr{I}_T$ (region inside the light brown dashed curve of Figure 4, right), of T. Although this method can avoid placing dense samples inside a triangle, it poses challenges. First, the approximation may lead to a smaller image than desired. For instance, in Figure 4 (right), $\mathscr{I}_T$ intersects with triangle T_4, T_5, T_7, but T' does not. Second, it may fail to produce an outer approximation when using a large τ under a highly curled field. In Figure 6, 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 this method, we will obtain a triangle T' (the black dash triangle) while the real image should be the light brown curved closure $\mathscr{I}_T$. A more rigorous method is needed.

A rigorous outer approximation of $f_\tau(\partial T)$ can be obtained by covering ∂T by squares of size ε and repeating the procedure in 1. This is less costly than directly computing $f_\tau(T)$ since the number of squares needed to cover ∂T is of order ε^{-1}. One still has to pay the cost of the rigorous enclosure integration methods which is higher than standard numerical methods for integrating a single initial point.

It should also be noted that if one uses an arbitrary τ-time discretization then f_τ may cease to be a homeomorphism. In this case this method fails to guarantee that the resulting image is an outer approximation. Thus, to use this method for the general τ-time discretization and to maintain rigor one needs to monitor that the image of ∂T remains a simple curve for all times up to time τ.

The following variant, though not rigorous, is reasonably safe as long as the image of ∂T remains a simple curve. It computes the outer approximation of the image of a triangle T.

1) Start from the two ending points of each edge, and trace them for the same small time $\nabla\tau$ ($\nabla\tau \ll \tau$), respectively
2) Compute the distance between the images of the two points. If it is larger than the threshold (e.g., half of the length of the original edge), compute the middle point between the two image points, then start tracing from it as well.
3) Repeat 2) until reaching the limit of time τ.
4) After completing the aforementioned steps, we obtain the approximate image of the edge. Repeat the same procedure to get the images of the other two edges of T. Computing the intersection of the underlying mesh with the closure formed by the images of the three edges gives rise to the outer approximation of the image of T.

This method still poses several difficulties. First, it is difficult to choose a proper $\nabla\tau$ to guarantee a small amount of advection of each sample. Second, monitoring the image of an edge incurs a high cost that makes it computationally prohibitive for large datasets and for datasets defined on curved surfaces that typically lack a global parameterization. For some applications, better performance is desired.

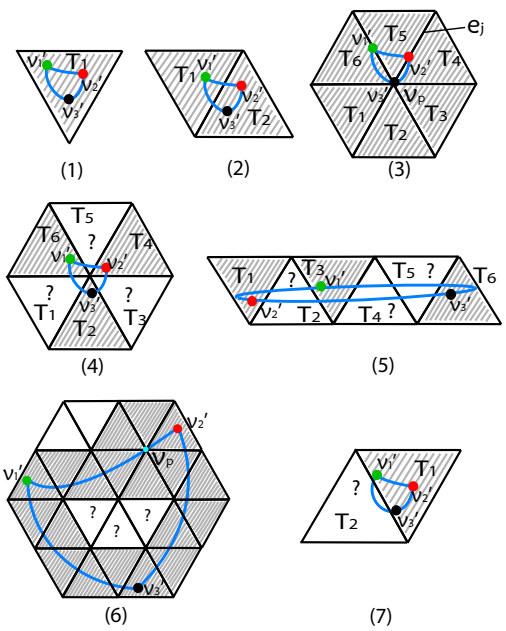

Fig. 7. Some possible cases of the image of a triangle under a flow.

B. Our Method For Computing Outer Approximations

In this section, we describe a method that can obtain enough information of the image $\mathscr{I}_T$ through the tracing of vertices and the heuristically chosen samples on the edges of the given mesh without having to compute the outer approximation explicitly. While this method is not rigorous in theory, it works for all the applications we have applied it to in practice.

Our method is based on the following observation: the image of a connected object under a continuous map is still connected. More specifically, the image of a triangle under a τ-map which is a continuous map is either a connected region, a simple curve or a point. And, the image of a line segment (e.g. an edge of the mesh) is a simple curve or a point. We now discuss our method in detail as follows.

We start with the study of some possible scenarios of the outer approximation of the image of a triangle T. Assume that T consists of three vertices v_1, v_2, v_3 and three edges e_1, e_2, e_3, where $e_1 = (v_2, v_3)$, $e_2 = (v_3, v_1)$ and $e_3 = (v_1, v_2)$. Considering the definition of an outer approximation in Section III-D, we let $I(T)$ represent the outer approximation of T obtained using certain numerical integration (such as, Runge-Kutta method). Similarly, let $I(v_i)$ represent the outer approximations of the images of the three vertices and $I(e_i)$ represent the outer approximations of the images of the three edges of T, respectively. Typically, $I(v_i)$ is a single triangle that contains the image of v_i if the integration error is smaller than the diameter of the triangle. To guarantee obtaining a sufficient outer approximation, if the image of v_i is located at a vertex v_p, we set $I(v_i)$ to be the one-ring neighborhood of v_p (Figure 7, cases (3) and (6)). If the image of v_i is located on an edge e_j, we set $I(v_i)$ to be the two triangles that have e_j as one common edge (Figure 7, case (3)).

Cases (1), (2) and (3) of Figure 7 show the first scenario. In this scenario, $I(T) \subseteq \cup_{i=1}^{3} I(v_i)$. That is, we only need to trace from the three vertices of T, the union of the outer approximations of them will give rise to the outer approximation of T.

Cases (4) and (5) of Figure 7 provide examples of Scenario 2.

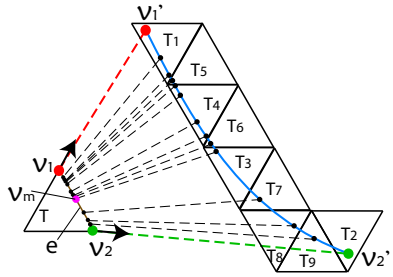

Fig. 8. This figure provides the notion of adaptive sampling on an edge $e(v_1v_2)$ (right). T is the original triangle. The image of edge v_1v_2 is $v_1'v_2'$. The dash lines show the mapping of the samples to the points on the image. The indexing of the triangles (right) indicate the order of computation.

In this scenario, $I(T) \subseteq \cup_{i=1}^{3} I(e_i)$. Therefore, the union of $I(v_i)$ will not provide us a sufficient outer approximation (for instance, triangles T_1, T_3, T_5 of case (4) in Figure 7 will be missing), but the union of $I(e_i)$ will. This requires us to keep track of the image of an edge. Section IV-A.2 presents an accurate solution, but it is computationally expensive. A more efficient method is desired.

Algorithm 1: Adaptive sampling on an edge

Routine: adaptive_edge_sampling(v_1, v_2, T_1, *original_T*, *neighbor_T*, V, X, τ, L)
Input: v_1,v_2: two vertices;
T_1, *original_T*, *neighbor_T*: triangles
V: vector field; X: surface; L: recursion level;
τ: user specified integral time
Output: the edges in the graph $\mathscr{F}_\tau$ related to the two triangles *original_T* and *neighbor_T*
Global variables: $\mathscr{F}_\tau$: the directed graph
Local variables: T_2: a triangle; s: a vertex
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*, $T_1, \mathscr{F}_\tau$);
new_edge(*neighbor_T*, $T_1, \mathscr{F}_\tau$);
return;
$T_2 \leftarrow$trace(v_2, τ);
if($T_1 == T_2$ || share_common_edge(T_1, T_2))
$v_1 \leftarrow v_2$; $T_1 \leftarrow T_2$;
new_edge(*original_T*, $T_1, \mathscr{F}_\tau$);
new_edge(*neighbor_T*, $T_1, \mathscr{F}_\tau$);
return;
else
$v_1 \leftarrow v_2$; $s \leftarrow v_2$;
$v_2 \leftarrow (v_1 + v_2)/2$;
call adaptive_edge_sampling(v_1, v_2, T_1, *original_T*, *neighbor_T*, V, X, τ, L);
$v_2 \leftarrow s$;
call adaptive_edge_sampling(v_1, v_2, T_1, *original_T*, *neighbor_T*, V, X, τ, L);
return;
End

Since we are interested in the outer approximation of an edge instead of the exact image of it, the connected triangle strip that contains the image of the edge is sufficient. The

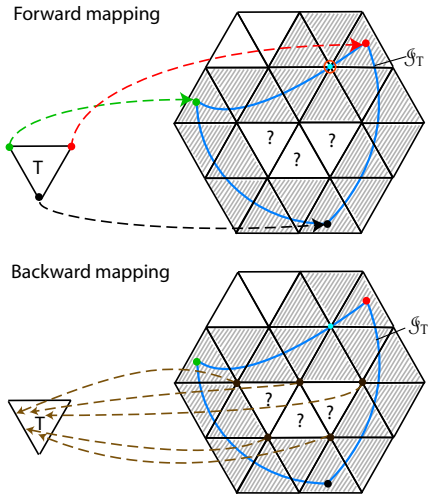

Fig. 9. A general example of the image of a triangle under a flow showing the scenario of case (6) in Figure 7. Through this example, we introduce the idea of backward mapping. The image in the top row illustrates the forward mapping. The red, green and black dashed curves indicate the forward mapping. Using adaptive edge sampling, we can find the connected triangle strip (the shaded region) that contains the image of the boundary of the triangle T. The bottom row image illustrates the idea of backward mapping. The interior vertices have been traced over the same time τ based on the inversed flow. The images of them will fall in the triangle T. The brown dashed curves indicate the backward mapping. Thus, we can obtain the remaining edges in the directed graph. Note that the boundary of the forward image $\mathscr{I}_T$ of T intersects with one vertex (highlighted by an orange circle). To obtain a sufficient outer approximation, we add the one-ring neighborhood of the vertex to the outer approximation.

connected triangle strip we refer to here is a triangle strip in which a pair of neighboring triangles share a common edge due to the aforementioned observation of the image of an edge under a continous map ($\tau-$ map here)(for example, Figure 8). We introduce the idea of adaptive edge sampling (Algorithm 1). The basic idea is that we first trace from the two vertices of an edge $e(v_1, v_2)$ (Figure 8, 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 do not process e further. Otherwise, more samples are then used until we obtain a connected triangle strip containing the image of e. To compute new samples, we make use of a binary search along the edge e . In detail, if the two triangles containing the images of the two vertices v_1, v_2 are neither the same nor neighbors, we then trace from the middle point v_m of the line segment (v_1, v_2) and determine whether the triangle T_3 that contains the image of v_m and the two triangles T_1 and T_2 form a connected triangle strip or not. If they are not, assume that among them T_1 and T_3 are not neighbors, It means that we need more samples on the line segment (v_1, v_m) to obtain the connected triangle strip between T_1 and T_3. Therefore, we compute the middle point of the line segment (v_1, v_m) and trace from it to obtain the triangle containing the image of it. Repeat this process until a connected triangle strip is found. Figure 8 demonstrates the idea of this algorithm. The indexing of the triangles indicate the order of computation. 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 τ, even though we sample densely along the edges. However, we have not experienced this problem in practice.

Using the adaptive edge sampling scheme, we successfully compute the outer approximation of scenario 2. But we will fail

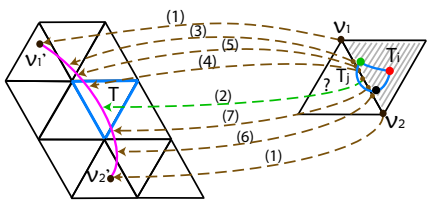

Fig. 10. This figure describes how the backward mapping and the adaptive edge sampling help to find the complete edges of the directed graph under a highly stretched flow. The edge (v_1v_2) has been sampled to obtain the continuous triangle strip that contains the image of it using Algorithm 1. The brown dashed curves illustrate the backward tracing along $-V$. (1)–(7) indicate the sampling and tracing order. Note that step (2) gives rise to the edge $T \to T_j$ that is missed in the case (7) of Figure 7.

under case (6) of Figure 7, which is an example of scenario 3. In this scenario, $I(T) \supset \cup_{i=1}^{3} I(e_i)$. Therefore, keeping track of the images of the three edges is not sufficient. More specificially, consider the image $\mathscr{I}_T$ of a triangle T under a flow V over time τ (Figure 9, top). In this case, we can find all the triangles that contain the images of the three edges of T using the adaptive edge sampling algorithm. But we are not able to find the interior triangles intersecting with $\mathscr{I}_T$. We observe that any sample inside T will be mapped to the image $\mathscr{I}_T$, and any sample inside $\mathscr{I}_T$ should be able to be mapped back to the interior of T as well (Figure 9, bottom row). That is, if we sample any point inside each inner triangle, and trace the sample point with respect to the inverse flow $-V$ over the same time τ, the image of it should fall in T. These observations motivate us to introduce the backward mapping as the complement of forward mapping when computing the outer approximation of the image of a triangle. Figure 9 (bottom row) illustrates the idea of the backward mapping. For the updating of the graph $\mathscr{F}$, if we trace backward from any sample of a triangle T_i over time τ, and its image falls in triangle T_j, we add an edge $T_j \to T_i$ to $\mathscr{F}$.

With the assist of backward tracing combined with the adaptive samping scheme, we now can compute a sufficient outer approximation for case (6) in Figure 7. Furthermore, more difficult case could be handled as well. Consider case (7) in Figure 7, $\mathscr{I}_T$ intersects with two triangles. Therefore, the outer approximation should include these two triangles, even though the images of the three vertices fall in the same triangle. Using both adaptive edge sampling and backward mapping, we can compute the outer approximation of this case correctly as follows. We first perform forward tracing, which will eventually generate an edge from triangle T to T_i. When we perform backward tracing, we first trace the two vertices of edge (v_1v_2) (step (1) of Figure 10) of the edge and determine whether the two triangles containing the images of the vertices of the edge are not neighbors. In here, they are not. We then choose the middle point of the edge and trace from it over time τ. It ends at triangle T. Therefore, we obtain the edge from T to T_j, since the edge (v_1, v_2) is shared by both T_i and T_j.

The logic of the complete algorithm is shown in Algorithm 2. We first trace each vertex v of a triangle T forward for the time τ. If it falls in triangle T_i, we add the edges from the triangles of the one-ring neighbors of v to T_i in $\mathscr{F}_\tau$. Second, we trace each vertex v of T backward with τ and find the triangle T_i' containing the advected vertex of v. We then add the edges from T_i' to the one-ring neighbors of v. Note that if the image of v is located at a vertex v' or on an edge e, T_i (or T_i') becomes a set of one-ring

triangle of v' or the two triangles sharing the edge e. Third, we compute the image of each edge following the original flow and inversed flow, respectively. The adaptive edge sampling algorithm is employed to produce an outer approximation in a practical and effective manner. The directed edges are added accordingly during the process. Note that the algorithm doesn't deal with the interior triangles of an image (Figure 9) explicitly, since the backward tracing using the same manner of forward tracing stage (i.e. proceed the vertices and edges, respectively) will eventually take care of those interior triangles.

Algorithm 2: An efficient outer approximation computation

Routine: construct_multivaluemap(V, X, τ, L)
Input: V: vector field; X: surface; τ: integral time
L: maximum recursion level
Output: $\mathscr{F}_\tau$: the completed graph
Local variables: T: current triangle; e : current edge;
N_T: the triangle sharing the edge e with T;
v_1,v_2: the two vertices of e
Begin
for each vertex v of X
$T \leftarrow$ trace_forward(v, τ);
new_edges(one-ring of v, T);
for each vertex v of X
$T \leftarrow$ trace_backward(v, τ);
new_edges(T, one-ring of v);
for each triangle T of X
for each edge e of T
if e is visited
continue;
else
$e \leftarrow$ visited;
$v_1, v_2 \leftarrow$ two vertices of e;
$N\_T \leftarrow$ the triangle sharing e with T;
/*forward mapping*/
call adaptive_edge_sampling(v_1, v_2, T, T,
N_T, V, X, τ, L);
/*backward mapping*/
call adaptive_edge_sampling(v_1, v_2, T, T,
N_T, $-V$, X, τ, L);
End

C. Result and Discussion

We have applied this algorithm to a number of analytical vector fields. Figure 2 provides the comparison of different Morse decompositions of a designed vector field using the geometry-based method (b) and the τ-maps with different time τ's (c) (d). The ECG of the vector field is shown (a). The corresponding MCGs of the obtained Morse decompositions (Figure 2(b-d)) are shown in the second row. From the results, we observe that the geometry-based mapping approach is fast ($0.14s$), but tends to result in a Morse decomposition that is too coarse (only four Morse sets have been extracted), while the MCG derived from an $\mathscr{F}_\tau$ has finer Morse decomposition (Figure 2(d), ten Morse sets have been found). Note that the MCG in (d) matches the ECG (a), although they are labeled differently. We also observe that the larger the τ, the finer the Morse decomposition is (i.e. closer to the optimal). Larger τ can provide more detailed information

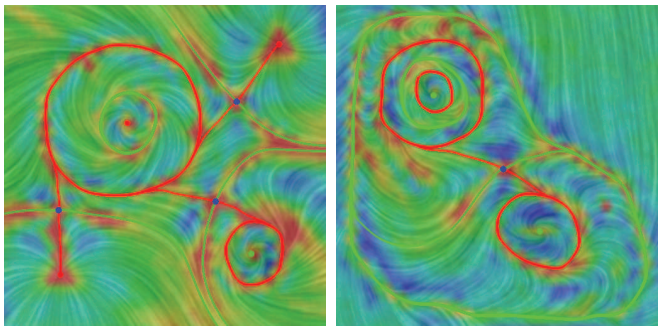

Fig. 11. This figure shows the density maps of the sample rates of the two analytic fields using our algorithm. The color coding of the density map uses red for the region with a larger sampling rate and blue for a lower sampling rate.

of the flow behavior. On the other hand, larger τ requires more computation time to construct MCGs, and larger integration errors may be introduced as well (Figure 6).

As is indicated in Figure 4, the τ-map approach leads to a combinatorial multi-valued map $\mathscr{F}$ with smaller images (than the geometry-based method) and hence a finer Morse decomposition. An important point that can easily be overlooked is the freedom of choice in the construction of $\mathscr{F}$. We have chosen an approach that is a compromise between accuracy of $\mathscr{F}$ 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 τ to refine the images. Alternatively, if one knows that the original vector field contains significant errors, and since the $\mathscr{F}$ needs only to be an outer approximation, these errors can be incorporated into the construction of the images of $\mathscr{F}$(Figure 5). Thus, even in the presence of considerable small perturbation (Figure 5) one can ascertain that the resulting MCG is valid.

An interesting observation is that to compute the sufficient outer approximation, our algorithm tends to use more samples for the flow regions with divergent behaviors (sources under forward mapping and sinks under backward mapping) and stretching behaviors (separatrices and periodic orbits). Figure 11 provides the density maps of sample rates of the two analytic fields using our algorithm. The color coding of the density map uses red for the region with a larger sampling rate and blue for a lower sampling rate. We observe that the regions having a larger sampling rate tend to coincide with the regions with highly stretched flow behavior. This verifies that our backward and forward mapping framework combined with adaptive sampling technique locates the flow regions with high distortion correctly.

V. Temporal τ vs. Spatial τ_s

The τ-map introduced previously refers to a time discretization, i.e., every particle travels for a time τ. We refer to it as a temporal τ-map. In many scientific data sets, the vector field magnitude of the underlying flow varies significantly. If a constant time τ is used, the advection of some triangles corresponding to the flow region with a slow speed may not be advected far enough in order to construct the edges of $\mathscr{F}_\tau$. One solution is to choose a τ that makes sure every triangle is advected sufficiently far. However, this is likely to affect the overall performance and introduce errors. Similar problems have appeared in texture- and streamline-based flow visualization. One popular approach is to normalize the vector field before generating the streamlines or

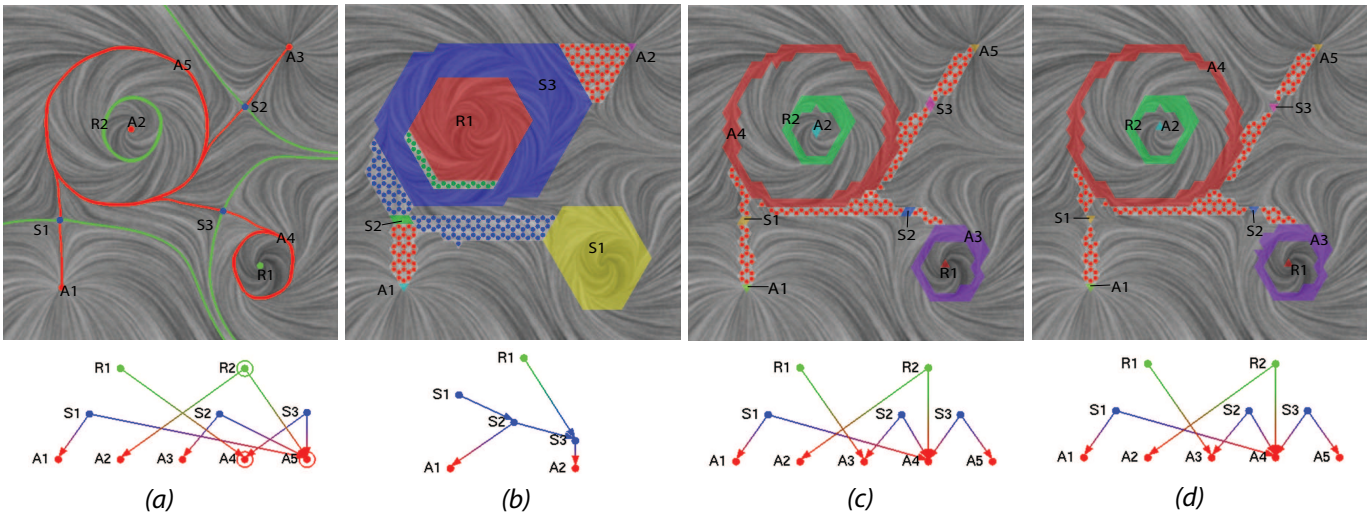

Fig. 12. This figure shows various analysis results of an analytical data set: (a) ECG, (b) MCG (geometry-based method), (c) MCG (temporal $\tau = 12$) and (d) MCG (spatial $\tau_s = 0.049$). The computational time for (b)-(d) is $0.17s$, $2.42s$ and $1.57s$, respectively. Notice how the Morse sets are refined by using the idea of τ-maps. We also observe that using a spatial τ_s-map for the analysis of this field can give rise to a comparable Morse decomposition (having the same Morse sets) to the one using a temporal τ with a faster performance. The visualization scheme of ECGs and MCGs are described in Figure 1.

advecting the textures. Under these normalized vector fields, the vector values at the vertices are scaled to have the same magnitude except for fixed points. Therefore, the streamline computation can be executed efficiently. Motivated by this observation, we propose the idea of a *spatial τ-map*, which we refer to τ_s-map.

More specifically, a τ_s-map is defined on a spatial discretization τ_s. When computing a τ_s-map in the computational domain (a triangle mesh X here), for each sample of the triangle T in X we keep track of the integral length of the sample following the flow until the accumulated integral length reaches the spatial constraint τ_s. Since all the particles will travel the same distance in the same speed (e.g. the maximum speed) everywhere except for the neighborhoods of the fixed points, one can expect a faster computation than tracing with respect to the original (non-normalized) vector field. When considering spatial τ_s, we still can reuse the framework in Algorithm 2 to compute the $\mathscr{F}_\tau$ with only difference being that we now accumulate integral length instead of integral time. One important concern is how to compute the correct trajectory when the tracing enters the neighborhoods of the fixed points. The basic rule is that the trajectory should not cross any fixed points. Fortunately, the flow will slow down in those neighborhoods according to the properties of fixed points (where vector magnitude equals zero) and the continuous approximation of the flow guaranteed by the interpolation schemes we are using (Section IV). Hence, we stop tracing when the vector magnitude is below a certain threshold (for instance, 0.01 times the uniform vector magnitude). We point out that after normalization, we have artificially introduced deviation to the original vector field.

We apply the idea of spatial τ_s to a designed vector field (Figure 12). The geometry domain of the vector field consists of 6144 triangles. Ten Morse sets have been extracted using a temporal $\tau = 12$. The extraction took 2.42 seconds on a 3.0 GHz PC with 1.0 GB RAM. With a spatial τ_s-map ($\tau_s = 0.049$), we extract the similar Morse sets using only 1.57 seconds. The result of the geometry-based method is also shown (Figure 12(b)). The corresponding MCGs and ECG of the field are also shown in the bottom row of Figure 12. Based on the results, we observe that using a spatial τ_s, we can achieve faster Morse decomposition

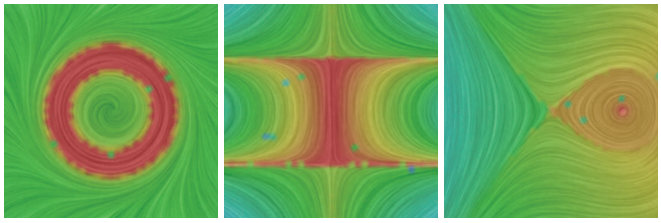

Fig. 13. The visualization of the integral time of different flow regions being spent under a constant spatial τ_s. An rainbow coloring scheme is used, where the red regions indicate larger tracing time is used and blue means smaller time.

(Figure 12(d)). The use of τ_s also extends our understanding of τ-maps. In the previous section, we set a constant τ for all flow regions during the $\mathscr{F}_\tau$ computation. It is not necessary and may lead to distortion of the outer approximation when large τ is used. The success of τ_s-maps shows that it is possible to use different τ's in different flow regions. This is because given a constant distance τ_s and different flow speed v_s, we will obtain different tracing time $t = \tau_s / v_s$ in different flow regions (Figure 13). Therefore, more heuristic information from the dynamics of the flow can be employed to guide the choice of a proper τ for a specific flow region. This is the challenge we plan to address in future research.

VI. Applications

In this section, we provide the vector field analysis results using the efficient Morse decomposition framework for two engine simulation data sets. They are the extrapolated boundary velocity fields that are obtained through simulation of in-cylinder flow. Engineers are interested in knowing whether or not the flows on the surface follow the ideal patterns [17].

Figure 14 shows the results of the gas engine simulation data. The first column shows the results using the geometry-based method. The second and third columns provide the results using the temporal τ-maps with $\tau = 0.1$ and $\tau = 0.3$, respectively. The corresponding MCGs are also displayed under the flow images. We observe that a Morse set has been extracted at the back of the chamber. It shows a recurrent pattern which indicates the flow

TABLE I

THE COMPLEXITY AND TIMING RESULTS FOR TWO CFD DATA SIMULATING IN-CYLINDER FLOW THROUGH A COMBUSTION ENGINE (FIGURES 14 AND 15). TIMES (IN SECONDS) ARE MEASURED ON A 3.6 GHZ PC WITH 3GB RAM.

dataset name(τ)	# polygons	# edges in $\mathscr{F}_\tau$	# Morse sets	time constructing $\mathscr{F}_\tau$	time extracting Morse sets	time computing MCG	time total
gas engine(temporal $\tau = 0.1$)	26,298	195,694	50	27.844	0.218	7.922	35.984
gas engine(temporal $\tau = 0.3$)	26,298	215,774	57	75.357	0.25	1.219	76.826
diesel engine(temporal $\tau = 0.3$)	221,574	2,035,133	200	1,101.323	7.781	37.703	1,146.807
diesel engine(spatial $\tau_s = 0.08$)	221,574	2,167,914	201	689.451	8.141	43.234	740.826

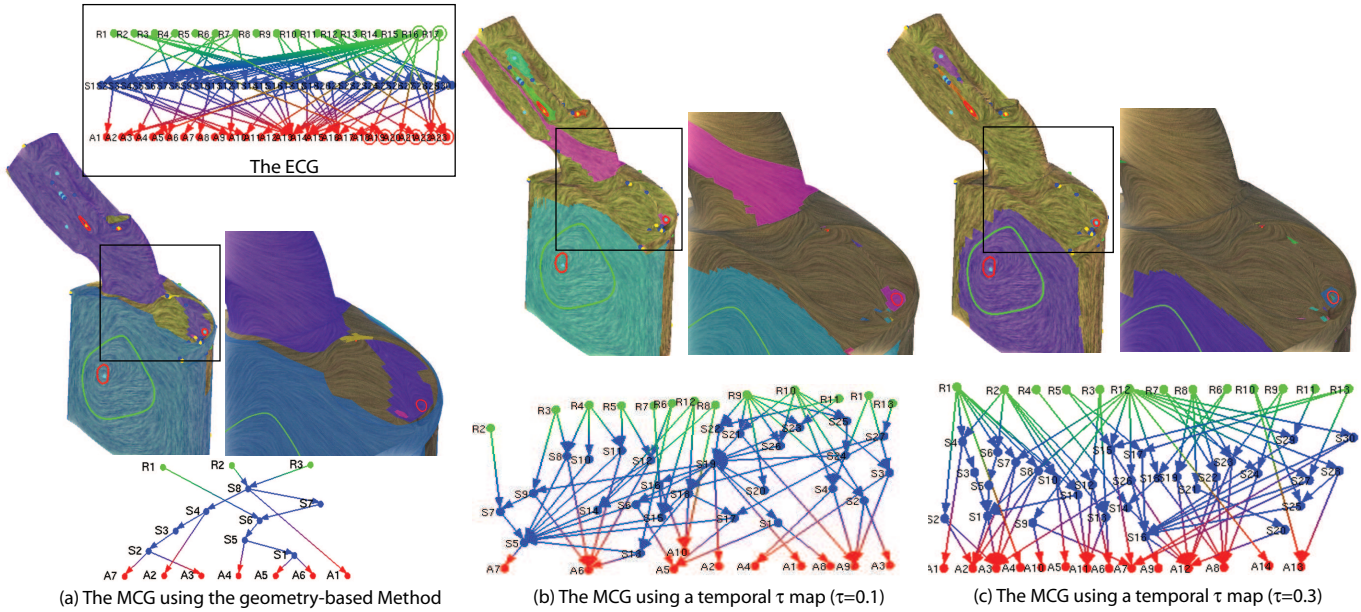

Fig. 14. This figure compares the results of the Morse decompositions of the gas engine simulation data obtained using geometry-based method (a), a temporal τ-map with $\tau = 0.1$ (b) and a temporal τ-map with $\tau = 0.3$ (c), respectively. Note that the color disk-like region at the back of the cylinder bounds the area of recirculating flow corresponding to tumble motion which indicates an ideal pattern of motion with good mixing properties. Notice that using the τ-maps can greatly improve the quality of the Morse decomposition (the zoom in images). The corresponding MCGs of different Morse decompositions and the ECG of the data are also shown.

starting to approximate the ideal tumble motion. The Morse sets obtained based on a τ-map capture regions that are more faithful to important features, while the approach using the geometry-based map could give rise to fewer Morse sets that cover large regions, which makes the identification of important features more difficult.

The results shown in Figure 15 are from the diesel engine simulation. The first column shows the results using the geometry-based method. Notice the rainbow-like regions indicate the recurrence behavior that does not actually exist. That is, the geometry-based method generates a Morse decomposition with misleading information. In the remaining columns, we provide two Morse decomposition results of the same data using a temporal τ-map ($\tau = 0.3$) and a spatial τ_s-map ($\tau_s = 0.08$), respectively. For the temporal case, the obtained Morse decomposition contains 200 Morse sets. It took $1,146.807$ seconds to obtain the result. For the spatial case, the number of the extracted Morse sets of the Morse decomposition is 201. The time for computing this Morse decomposition is 740.826 seconds. Either temporal τ method or spatial τ_s method provides accurate information of the recurrence behavior of the bottom of the in-cylinder of the diesel engine, but the spatial τ_s-map shows faster $\mathscr{F}_\tau$ computation than temporal τ-map scheme.

Table 1 provides the performance information of the two data sets using different $\mathscr{F}_\tau$s.

VII. CONCLUSION

In this paper, we have demonstrated the fundamental difficulties associated with the definition of vector field topology based on individual trajectories. As a solution, we advocate the use of a Morse connection graph to represent the topology of a vector field. Moreover, we have described an efficient framework for computing Morse decompositions of vector fields. 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. This makes it more suitable for a rigorous interpretation of vector field topology. To obtain a finer MCG than previous method (i.e. the geometry-based method), we employ the idea of τ-maps to perform flow combinatorialization and encode the flow dynamics into a directed graph $\mathscr{F}_\tau$, upon which we perform Morse decomposition. In order to compute $\mathscr{F}_\tau$ efficiently, we make use of both forward and backward tracing and introduce an adaptive sampling algorithm along the edges to account for the discontinuity problem while computing the

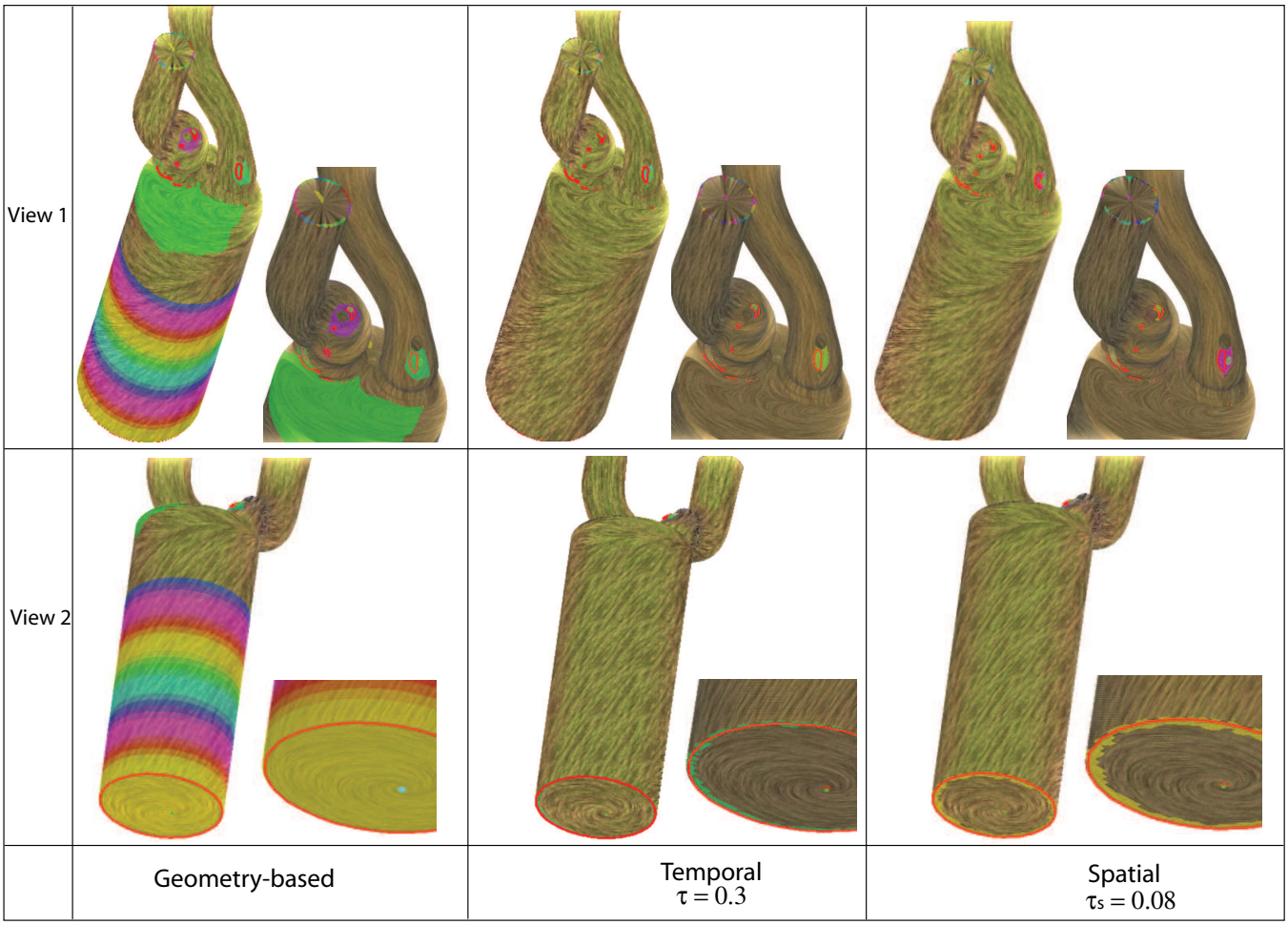

Fig. 15. A comparison of various Morse decompositions of the diesel engine simulation data set. The first column shows the Morse neighborhoods obtained using the geometry based mapping. The color rainbow-like regions indicate the possible recurrent flow behavior. The second column provides the results using a temporal τ-map with $\tau = 0.3$, while the third column gives the results using a spatial τ_s map with $\tau_s = 0.08$. Note how much more refined the topological regions become. We also observe that using a proper spatial τ_s, we can obtain comparable Morse decomposition with higher performance (See Table 1).

approximate image. As an alternative to the temporal τ-map, we present the use of a spatial τ_s-map, which typically provides faster computation than temporal τ-map scheme with similar fineness in the MCGs. We show the utility of our approach in a number of applications including analytical data and two engine simulation data sets on surfaces. We should point out that the limitation of our approach is that the optimal Morse decomposition of a vector field under a given mesh is constrained by the resolution of the mesh. This can potentially be fixed by using other graphics techniques, such as flow-guided remeshing or subdivision. Our framework allows engineers to choose between individual trajectory-based topology analysis given its instability and relatively stable analysis, or choose between higher performance and higher quality when applying the more stable analysis.

There are a number of future directions. First, MCGs have the potential of enabling a multi-scale representation of the flow, which can be used to guide vector field clustering, vector field compression and automatic simplification. Second, exploration of τ magnitude in different regions of the domain is important. Third, exploring a discrete integration scheme to compute the discrete function (i.e., the multi-valued map, $\mathscr{F}$), may further improve the performance of our method as well. Finally, there is a need to extend the work to time-dependent vector fields.

ACKNOWLEDGMENT

The authors would like to thank our anonymous reviewers for their valuable comments and suggestions. This work was partially funded by US National Science Foundation (NSF) Grant CCF-0546881. Konstantin Mischaikow was partially supported by NSF DMS 0511115, and grants from D.O.E. and DARPA.

REFERENCES

[1] Computer assisted proofs in dynamics group. http://capd.wsb-nlu.edu.pl/.
[2] U. M. Ascher and L. R. Petzold. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1998.
[3] E. Boczko, W. Kalies, and K. Mischaikow. Polygonal approximation of flows. *Topology and its Applications*, 154(13):2501–2520, 2006.
[4] 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(4):769–785, Jul./Aug. 2007.
[5] C. Conley. Isolated invariant sets and the morse index. *CMBS Regional Conference Series in Mathematics*, 38, 1978.
[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to algorithms*. MIT Press, Cambridge, MA, 1990.
[7] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. In *Proceedings of IEEE Visualization 94*, pages 140–147, 1994.

[8] M. Eidenschink. *Exploring Global Dynamics: A Numerical Algorithm Based on the Conley Index Theory*. PhD thesis, Georgia Institute of Technology, 1996.
[9] J. Hale and H. Kocak. *Dynamics and Bifurcations*. New York: Springer-Verlag, 1991.
[10] 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.
[11] J. L. Helman and L. Hesselink. Visualizing Vector Field Topology in Fluid Flows. *IEEE Computer Graphics and Applications*, 11(3):36–46, May 1991.
[12] C. Johnson. Top Scientific Visualization Research Problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, July/August 2004.
[13] W. Kalies and H. Ban. A computational approach to conley's decomposition theorem. *Journal of Computational and Nonlinear Dynamics*, 1(4):312–319, 2006.
[14] 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.
[15] R. Laramee, H. Hauser, L. Zhao, and F. H. Post. Topology Based Flow Visualization: The State of the Art. In *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)*, Mathematics and Visualization, pages 1–19. Springer, 2007.
[16] 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 of IEEE Visualization 2005*, pages 623–630, 2005.
[17] R. S. Laramee, D. Weiskopf, J. Schneider, and H. Hauser. Investigating Swirl and Tumble Flow with a Comparison of Visualization Techniques. In *Proceedings of IEEE Visualization 2004*, pages 51–58, 2004.
[18] K. Mischaikow. Topological techniques for efficient rigorous computation in dynamics. *Acta Numer.*, 11:435–477, 2002.
[19] K. Mischaikow and M. Mrozek. Conley index. In *Handbook of dynamical systems, Vol. 2*, pages 393–460. North-Holland, Amsterdam, 2002.
[20] M. Mrozek and P. Z. Nski. Set arithmetic and the enclosing problem in dynamics. *Annales Pol. Math.*, 74:237–259, 2000.
[21] 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.
[22] 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.
[23] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.
[24] G. Scheuermann, H. Hagen, H. Krüger, M. Menzel, and A. Rockwood. Visualization of Higher Order Singularities in Vector Fields. In *Proceedings of IEEE Visualization '97*, pages 67–74, Oct. 1997.
[25] 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.
[26] G. Sheuermann, X. Tricoche, and H. Hagen. C1-interpolation for vector field topology visualization. In *Proceedings of IEEE Visualization '99*, pages 271–278, 1999.
[27] J. Stam. Flows on surfaces of arbitrary topology. In *ACM Transactions on Graphics (SIGGRAPH 03)*, volume 22, pages 724–731, July 2003.
[28] H. Theisel, T. Weinkauf, 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.
[29] 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.
[30] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. In *Proceedings of IEEE Visualization 01*, pages 159–166, 2001.
[31] G. Turk. Texture synthesis on surfaces. In *Proceedings of ACM SIGGRAPH 01*, pages 347–354, 2001.
[32] J. J. van Wijk. Image based flow visualization. In *ACM Transactions on Graphics (SIGGRAPH 02)*, volume 21, pages 745–754, Jul 2002.
[33] L.-Y. Wei and M. Levoy. Texture synthesis over arbitrary manifold surfaces. In *Proceedings of ACM SIGGRAPH 01*, pages 355–360, 2001.
[34] T. Wischgoll and G. Scheuermann. Detection and Visualization of Closed Streamlines in Planar Fields. *IEEE Transactions on Visualization and Computer Graphics*, 7(2):165–172, 2001.
[35] 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.
[36] 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. 2001.
[37] 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.
[38] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25(4):1294–1326, 2006.

Guoning Chen received the BE degree from Xian Jiaotong University in 1999 and the ME degree from Guangxi University, China, in 2002. He is pursuing the PhD degree in computer science at Oregon State University. He is currently working with Eugene Zhang on vector field analysis, visualization, and design. He is also interested in tensor field design and its applications in computer graphics.

Konstantin Mischaikow received the BA and MA degrees in mathematics from Reed College in 1979 and 1983, respectively, and the PhD degree in mathematics from the University of Wisconsin-Madison in 1985. He was a lecturer of the University of Minnesota-Minneapolis from 1984 to 1985. From 1985 to 1987, he was working in Brown University as a visiting assistant professor. He was an assistant professor from 1987 to 1990 and an associate professor from 1990 to 1991 at Michigan State University. He was an assistant professor from 1989 to 1991 and an associate professor from 1991 to 1996 at Georgia Tech. From 1996 to 2006, he was a professor of Georgia Tech. He is now a professor at Rutgers University.

Robert S. Laramee received a bachelors degree in physics, cumlaude, from the University of Massachusetts, Amherst in 1997. In 2000, he received a masters degree in computer science from the University of New Hampshire, Durham. He was awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics in 2005. From 2001 to 2006 he was a researcher at the VRVis Research Center (www.vrvis.at) and a software engineer at AVL (www.avl.com) in the de- partment of Advanced Simulation Technologies. Currently he is a Lecturer (Assistant Professor) at the Swansea University (Prifysgol Cymru Abertawe), Wales in the Department of Computer Science (Adran Gwyddor Cyfrifiadur). His research interests are in the areas of scientific visualization, computer graphics, and human-computer interaction. He is a member of the IEEE Computer Society.

Eugene Zhang received the PhD degree in computer science in 2004 from the Georgia Institute of Technology. He is currently an assistant professor at Oregon State University, where he is a member of the School of Electrical Engineering and Computer Science. His research interests include computer graphics, scientific visualization, geometric modeling and processing, and computataional topology. He is a member of the IEEE Computer Society.