Vector Field Editing and Periodic Orbit Extraction Using Morse Decomposition

Guoning Chen, Konstantin Mischaikow, Robert S. Laramee, *Member, IEEE,* Paweł Pilarczyk, and Eugene Zhang, *Member, IEEE,*

***Abstract*— Design and control of vector fields is critical for many visualization and graphics tasks such as vector field visualization, fluid simulation, and texture synthesis. The fundamental qualitative structures associated with vector fields are fixed points, periodic orbits, and separatrices. In this paper we provide a new technique that allows for the systematic creation and cancellation of fixed points and periodic orbits. This technique enables vector field design and editing on the plane and surfaces with desired qualitative properties.**

The technique is based on *Conley theory* which provides a unified framework that supports the cancellation of fixed points and periodic orbits. We also introduce a novel periodic orbit extraction and visualization algorithm that detects, for the first time, periodic orbits on surfaces. Furthermore, we describe the application of our periodic orbit detection and vector field simplification algorithm to engine simulation data demonstrating the utility of the approach.

We apply our design system to vector field visualization by creating datasets containing periodic orbits. This helps us understand the effectiveness of existing visualization techniques. Finally, we propose a new streamline-based technique that allows vector field topology to be easily identified.

***Index Terms*— Vector field design, vector field visualization, vector field topology, vector field simplification, Morse decomposition, Conley index, periodic orbit detection, connection graphs.**

I. Introduction

VECTOR fields arise as models in almost all scientific and engineering endeavors which involve systems that change continuously. In the case of two-dimensional systems that can be modelled by vector fields defined on surfaces, visualization can play an important role in understanding the essential features in the system. This is also true for two-dimensional vector fields that are linked to potentially noisy data, such as a velocity field extracted from experiments or numerical simulations of fluids. In both cases, there are occasions in which one wishes to simplify the dynamic structure in a coherent admissible manner [1]. This latter step requires the ability to edit the underlying vector field. Furthermore, there are problems where the construction and modification of a vector field represents a preliminary step towards a larger goal such as texture synthesis [2], [3], [4] and fluid simulation for special effects [5].

There is substantial literature on the subject of vector field topology extraction and simplification, with considerable focus on the identification and manipulation of *fixed points* (see [6] and references therein). On the other hand, *periodic orbits* are essential structures of *non-gradient* vector fields, such as those in electromagnetism, chemical reactions, fluid dynamics, locomotion control, population modelling, and economics. There is a fundamental need to be able to incorporate them into the subject of vector field visualization and design. For example, Figure 1 shows the swirl motion of fluid in a combustion chamber using simulation [7]. Periodic orbits appear in some planar slices along the main axis of the chamber (middle) as well as the boundary geometry (Figure 2). The existence and locations of the periodic orbits provide clues to the swirl motion inside the chamber. Efficient periodic orbit detection and vector field visualization can help design engineers better understand how the shape of the chamber and the initial speed of the fluid through the intake ports impact engine efficiency.

Many of the aforementioned applications involve systems of *nonlinear* ordinary differential equations, for which explicit analytic solutions do not exist. The lack of analytic expressions led to the development of the subject of dynamical systems where the focus is on the qualitative structure of solutions. In the case of two-dimensional vector fields, the classical theoretical description of the dynamics is based on identifying fundamental topological and geometric structures such as fixed points, periodic orbits, separatrices, and their relationships [8], [9]. However, in practice there are at least two essential difficulties with this approach. First, unambiguously identifying all the topological structures for a vector field is impossible. Second, the existence of noise reduces the importance of objects such as fixed points and periodic orbits.

In this paper, we develop a vector field visualization and design system that extracts and visualizes boundary flow topology. This includes analyzing and modifying the vector field. The system builds on the ideas presented by Zhang et al. [6], and it provides the user with a variety of capabilities in that fixed points, periodic orbits, and separatrices can be identified.

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

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

Paweł Pilarczyk is with the Department of Mathematics, Kyoto University, Kyoto 606-8502, Japan.

Email:pawelpil@math.kyoto-u.ac.jp

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

Email:zhange@eecs.oregonstate.edu.

Manuscript received August 23, 2006; revised November 23, 2006; accepted December 4, 2006.

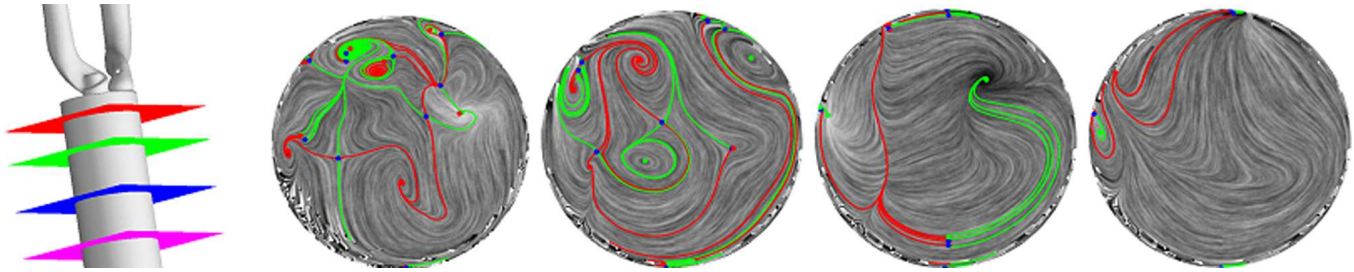


Fig. 1. Visualizing the simulation of flow in a diesel engine: the combustion chamber (leftmost) and four planar slices of the flow inside the chamber for which the plane normals are along the main axis of the chamber. From left to right are slices cut at 10%, 25%, 50%, and 75% of the length of the cylinder from the top where the intake ports meet the chamber. The vector fields are defined as zeros on the boundary of the geometry (*no-slip condition*). The automatic extraction and visualization of flow topology allows the engineer to gain insight into where the ideal pattern of swirl motion is realized inside the combustion chamber. In fact, the behavior of the flow and its associated topology, including periodic orbits, is much more complicated than the ideal. Figure 2 provides complementary visualization of the flow on the boundary of the diesel engine.

Furthermore, fixed points and periodic orbits can be created and removed subject to inherent topological constraints. To address the additional complexity dealing with periodic orbits, we make the following contributions in the presented research:

1) We provide a general framework and efficient algorithms that allow topological simplification on arbitrary vector fields defined on surfaces (Section VI). Our framework is based on Conley theory, which is a well-known theory in *non-linear dynamics* dating back the early 1970's (a brief introduction is available in Section III). To our knowledge, previous work including Zhang et al. [6] does not address simplification that involve periodic orbits. In addition, most of the existing simplification algorithms require planar vector fields.
2) We describe a novel graph-based representation of a vector field based on Morse decomposition, which we refer to as *Morse Connection Graphs* (*MCG*). This graph contains supplementary information with respect to the well-known *vector field skeleton* in that it addresses periodic orbits. We also provide an algorithm to efficiently compute MCG as well as their refinement (*Entity Connection Graphics*, or *ECG*) (Section V-B).
3) Our system allows a user to create periodic orbits on surfaces (Section IV). To do so, we combine the ideas of basis vector fields and constraint optimization. To our knowledge, this is the first time a periodic orbit creation algorithm is proposed and implemented.
4) As part of MCG and ECG construction, we present a novel and practical algorithm for periodic orbit extraction without first having to compute separatrices (Section V-A). Our method is based on the topological and geometric analysis of a vector field, and it enables extraction of periodic orbits–even those that are not accessible via fixed points.
5) The utility of our topological analysis, including periodic orbit detection and vector field simplification, is demonstrated in the context of a novel application, namely, the visualization of in-cylinder flow from automotive engine simulation data (Sections V and VI). Our algorithm for periodic orbit detection and ECG construction only takes less than a minute on one such dataset with nearly $900,000$ triangles.
6) We propose an enhanced streamline-based method in which periodic orbits and separatrices are highlighted (Section VII). This is particularly desirable for vector fields on surfaces since only portions of a periodic orbit may be visible for any given viewpoint (Figure 7).

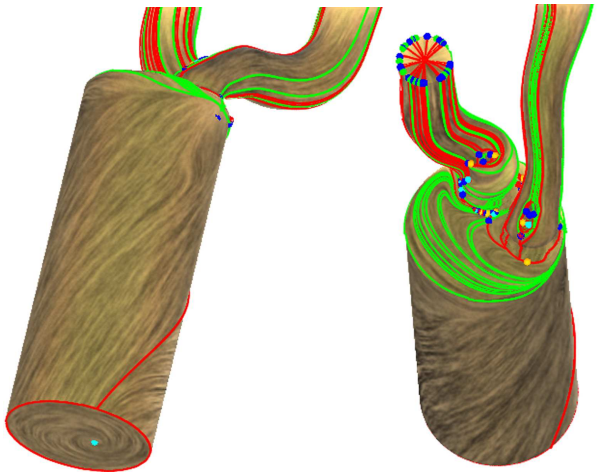


Fig. 2. The visualization of CFD data simulating in-cylinder flow through a diesel engine from two viewpoints. Compare them to the idealized flow shown in Figure 13 (left). Figure 1 provides complementary visualization of the flow inside the diesel engine. Both the texture and the topology-based visualizations indicate a nice pattern of swirl motion at the boundary of the combustion chamber while the regions near the intake ports reveal deviation from the ideal.

Because of the essential mathematical difficulties mentioned earlier, our numerical methods do not focus directly on fixed points, periodic orbits, and separatrices. Rather, we employ techniques based on Conley's purely topological approach to dynamical systems [10]. Broadly speaking, our approach is based on three steps. The first is to identify regions on which the dynamics exhibits recurrent behavior, i.e. fixed points and periodic orbits, and/or gradient-like behavior, i.e. separatrices. This involves the construction of *Morse decompositions*. A theoretical computational foundation for the types of algorithms we employ can be found in [11], [12]. The second is to identify the type of dynamics occuring in these regions, i.e. the existence of fixed points, periodic orbits, and separatrices. This is done using numerical methods and the *Conley index*. It should be noted that the Conley index not only generalizes the Poincaré index as it applies to fixed points, but it also provides information about the existence of periodic orbits. Finally, the vector field is modified in the identified regions to produce the desired dynamics.

The rest of the paper is organized as follows: Section II provides a brief review of related work on topology-based vector field visualization. Section III reviews related work and introduces Conley theory. Section IV introduces our method for creating periodic orbits on the plane and surfaces. Section V describes our periodic orbit detection technique and provides an algorithm for the construction of the MCG and ECG of a vector field. A general framework for various cancelling operations is presented in Section VI. Section VII provides details on our enhanced streamline based flow visualization technique followed by a discussion of possible future work in Section VIII.

II. Related Work

Vector field visualization, analysis, simplification, and design have received much attention from the Visualization community over the past twenty years. Much excellent work exists, and to review it all is beyond the scope of this paper. Here, we only refer to the most relevant work. Interested readers can find a complete survey in [13], [14], [15].

A. Vector Field Design

There has been some work in creating vector fields on the plane and surfaces, most of which is for graphics applications such as texture synthesis [2], [3], [4] and fluid simulation [5]. These methods do not address vector field topology, such as fixed points. There are a few vector field design systems that make use of topological information. For instance, Rockwood and Bunderwala [16] use ideas from geometric algebra to create vector fields with desired fixed points. Van Wijk [17] develops a vector field design system to demonstrate his image-based flow visualization technique (IBFV). The basic idea of this system is the use of basis vector fields that correspond to various types of fixed points. This system is later extended to surfaces [18], [19]. None of these methods provide explicit control over the number and location of fixed points since unspecified fixed points may appear. Theisel [20] proposes a planar vector field design system in which the user has complete control over fixed points and separatrices. However, this requires the user to provide the complete *topological skeleton* of the vector field, which can be labor-intensive. Recently, Zhang et al. [6] develop a design system for both planar domains and surfaces. This system provides explicit control over the number and location of fixed points through *fixed point pair cancellation* and *movement* operations. Our work is inspired by their system. However, we enable automatic extraction and visualization of periodic orbits on surfaces. We also introduce topology simplification operations for periodic orbits. There has also been recent work by Weinkauf et al. [21] on the design of 3D vector fields.

B. Vector Field Topology and Analysis

Helman and Hesselink [22] introduce vector field topology for the visualization of vector fields. They also propose efficient algorithms to extract vector field topology. Following their footsteps, much research has been conducted in topological analysis of vector fields. For example, Scheuermann et al. [23] use clifford algebra to study the non-linear fixed points of a vector field and propose an efficient algorithm to merge nearby first-order fixed points. Tricoche et al. [1] and Polthier and Preuβ [24] give efficient methods to locate fixed points in a vector field. Wischgoll and Scheuermann [25] develop a method to extract closed streamlines in a 2D vector field defined on a triangle mesh. Note that closed streamlines are in fact attracting and repelling periodic orbits. Theisel et al. [26] propose a mesh-independent periodic orbit detection method for planar domains. In contrast to these approaches, our automatic detection algorithm is extended to surfaces. Furthermore, this is the first time periodic orbit extraction and visualization has found utility in a real application.

C. Vector Field Simplification

Vector field simplification refers to reducing the complexity of a vector field. There are two classes of simplification techniques: topology-based (TB), and non-topology-based (NTB) [6]. Existing NTB techniques are usually based on performing Laplacian smoothing on the potential of a vector field inside the specified region. One example of these work is by Tong et al. [27], who decompose a vector field using Hodge-decomposition and then smooth each-component independently before summing them.

TB techniques simplify the topology of a vector field explicitly. Tricoche et al. [1] simplify a planar vector field by performing a sequence of cancelling operations on fixed point pairs that are connected by a separatrix. They refer to this operation as *pair annihilation*. A similar operation, named *pair cancellation*, has been used to remove a wedge and trisector pair in a tensor field [28]. Edelsbrunner et al. [29] perform pair cancellation on scalar fields defined on surfaces by changing the values of the scalar function near the fixed point pair. This is equivalent to simplifying the gradient vector field of the scalar function. We will follow this convention and refer to such an operation as fixed point pair cancellation. Zhang et al. [6] provide a fixed point pair cancellation method based on Conley theory. They also extend this operation to surfaces and to fixed point pairs that are *not* connected by a separatrix, such as a center and saddle pair. In this paper, we describe a more general framework for cancelling object pairs such as fixed points and periodic orbits (Section VI).

III. Background on Vector Fields

Our control of vector fields on surfaces is done using concepts from the topological theory of dynamical systems. Consider a manifold M and a subset $X \subset M$. The boundary of X is denoted by ∂X and closure by $\mathrm{cl}(X)$.

Mathematically, a vector field can be expressed in terms of a differential equation $\dot{x} = V(x)$. 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. Other

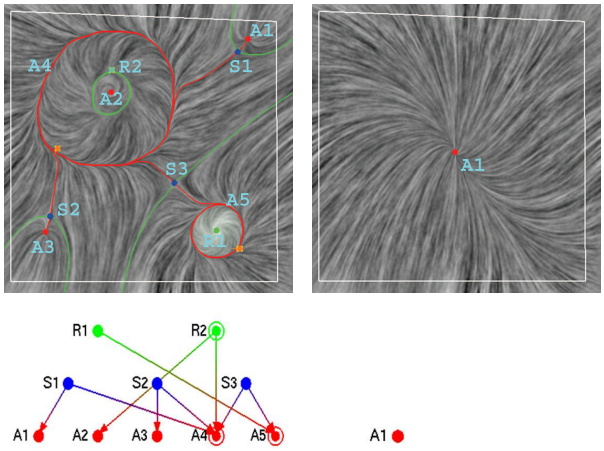


Fig. 3. An example vector field (upper left) and its ECG (lower left). The vector field contains a source (green), three sinks (red), three saddles (blue), a repelling periodic orbit (green), and two attracting periodic orbits (red). Separatrices that connect a saddle to a repeller (a source or a periodic orbit) are colored in green, and to an attractor (a sink or a periodic orbit) are colored in red. The fixed points and periodic orbits are the nodes in the ECG (lower left) and separatrices are the edges. In addition, a periodic orbit can be connected directly to a source, sink, or another periodic orbit. Such connections are also depicted as edges in the ECG. The simplified field of (upper left) is shown in (upper right) and its corresponding ECG is (lower right). Notice the Conley index for both vector fields inside the white loop are the same, which allows the vector field in the left to be simplified into the one shown in the right.

simple examples of invariant sets include the following. 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*.

Consideration of the important qualitative structures associated with vector fields on a surface requires familiarity with hyperbolic fixed points, periodic orbits and separatrices. Let x_0 be a fixed point of a vector field $\dot{x} = V(x)$; that is $V(x_0) = 0$. The linearization of V about x_0, results in a 2×2 matrix $Df(x_0)$ which has two (potentially complex) eigenvalues $\sigma_1 + i\mu_1$ and $\sigma_2 + i\mu_2$. If $\sigma_1 \neq 0 \neq \sigma_2$, then x_0 is called a *hyperbolic fixed point*. Observe that on a surface there are three types of hyperbolic fixed points: *sinks* $\sigma_1, \sigma_2 < 0$, *saddles* $\sigma_1 < 0 < \sigma_2$, and *sources* $0 < \sigma_1, \sigma_2$. 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. A periodic orbit Γ is *attracting* if there exists $\varepsilon > 0$ such that for every x which lies within a distance ε of Γ, $\omega(x) = \Gamma$. A *repelling* periodic orbit can be similarly defined ($\alpha(x) = \Gamma$)). Finally, given a point $x_0 \in M$, its trajectory is a *separatrix* if the pair of limit sets $(\alpha(x), \omega(x))$ consist of a saddle fixed point and another object that can be a source, a sink, or a periodic orbit. Figure 3 provides an example vector field (upper-left). Fixed points are highlighted by colored dots (sources: green; sinks: red; saddles: blue). Periodic orbits are colored in green if repelling and in red if attracting. Separatrices that terminate in a source or a repelling periodic orbit are shown in green and those terminate in a sink or an attracting periodic orbit are colored in red. For convenience, we will refer to a source and a sink as a *node* in the remainder of the paper wherever appropriate.

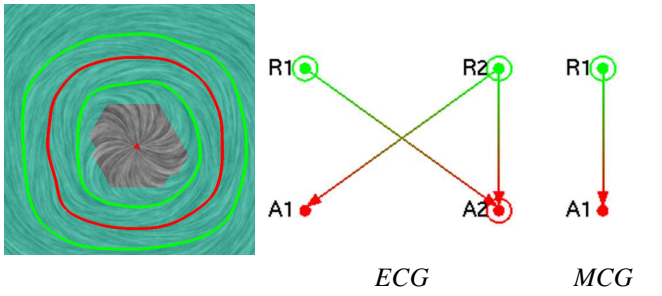


Fig. 4. This example shows the difference between ECG and MCG for a piecewise linear vector field created using our tool. The vector field shown in the left contains one fixed point and three periodic orbits. Therefore, the ECG consists of four nodes (middle). However, due to the resolution of the underlying mesh, there are only two Morse sets (colored regions) with one containing the fixed point and the other containing the periodic orbits. Consequently, there are two nodes in the MCG (right).

Even for flows restricted to surfaces, invariant sets can be extremely complicated and cannot be assumed to consist of hyperbolic fixed points, periodic orbits and separatrices [30]. Furthermore, even if the recurrent dynamics is restricted to fixed points and periodic orbits, it is impossible to develop an algorithm that will identify all of them. For example, it is easy to generate continuous vector fields that contain infinitely many isolated fixed points and/or periodic orbits. Even for continuous piecewise linear vector fields, it is not clear whether infinitely many isolated periodic orbits may exist. Further investigation is required to answer this question. Thus we require a language that allows us to manipulate a broader but useful class of invariant sets.

A. Morse Decomposition and Connection Graphs

A compact set $N \subset M$ is an *isolating neighborhood* if for all $x \in \partial N$, $\varphi(\mathbf{R},x) \not\subset N$. That is, the flow enters or leaves N eventually everywhere on ∂N. An invariant set S is *isolated* if there exists an isolating neighborhood N such that S is the maximal invariant set contained in N. Observe that hyperbolic fixed points and periodic orbits are examples of isolated invariant sets. Isolated invariant sets posses two essential properties. First, there are efficient algorithms for identifying isolating neighborhoods [11]. Second, there exists an index, called the Conley index [31], that identifies the types of modifications to the structure of the invariant set that are topologically permissible. For example, the Conley index of the vector field shown in Figure 3 (upper-left) inside the white loop is identical to that of a sink. Topological simplification of the complex field inside the region can result in the field shown in the right.

Central to our effort is the need for a computationally robust decomposition of invariant sets. A *Morse decomposition*, $\mathscr{M}(S)$, of S consists of a finite collection of isolated invariant subsets of S, called *Morse sets*,

$$\mathscr{M}(S) := \{M(p) \mid p \in \mathscr{P}\} \tag{3}$$

such that if $x \in S$, then there exists $p,q \in \mathscr{P}$ such that $\alpha(x) \subset M(q)$ and $\omega(x) \subset M(p)$. Furthermore, there exists a partial order $>$ on $\mathscr{P}$ satisfying $q > p$ if there exists $x \in S$ such that $\alpha(x) \subset M(q)$ and $\omega(x) \subset M(p)$. Let $C(p,q) := \{x \in M \mid \alpha(x) \subset M(p) \text{ and } \omega(x) \subset M(q)\}$. An efficient means

of presenting the partial order on a Morse decomposition is given by the associated *Morse Connection Graph* (MCG) which is the minimal directed graph whose vertices consist of the Morse sets $\{M(p) \mid p \in \mathscr{P}\}$ and whose directed edges $M(q) \to M(p)$ imply $q > p$. Figure 3 (lower-left) shows an MCG of the vector field in the upper-left. Here $\mathscr{P}$ is the set of labels ($R1$ and $R2$, $S1$-$S3$, and $A1$-$A5$), and $M(p)$ is the actual object that p represents, i.e., $M(R1)$ is a source. Note that a MCG contains supplementary information with respect to the topological skeleton presented by Helmann and Hesselink [22]. For example, consider the idealized magnetic field over the Earth's surface in which only two fixed points exist and none of the connecting orbits between them is a separatrix. Similarly, a periodic orbit can be connected to a source (Figure 7, left) or another periodic orbit (Figure 7, middle) without any separatrices in the field.

Computing a Morse decomposition and its associated MCG can be done as follows. Let $\mathscr{T}$ denote a triangulation of the phase space. An edge in this triangulation is classified as a *transverse edge* if the flow leaves one of incident triangles completely (a one-way road). Otherwise, the edge is *nontransverse* (two-way). Construct equivalence classes on $\mathscr{T}$ using the following relationship and transitivity. Two triangles $T_0, T_1 \in \mathscr{T}$ are equivalent if $T_0 \cap T_1$ consists of a nontransverse edge. Taking the union of all triangles in an equivalence class produces a polygonal region, who boundary consists of transverse edges only. Let $\mathscr{R}$ denote the resulting collection of polygons which tile the phase space. Define a directed graph whose vertices consist of the polygons in $\mathscr{R}$. Given $R_0, R_1 \in \mathscr{R}$, there exists an edge from R_0 to R_1 if and only if $R_0 \cap R_1$ contains an edge and the vector field points from R_0 to R_1 along the edge. Finally, fix a tolerance $\varepsilon > 0$ (to avoid missing fixed points during extraction) and if there exists a point in R_0 at which $\|V\| \leq \varepsilon$ then include an edge from R_0 to itself. It is proven in [11] that the maximal invariant sets within the strongly connected path components of this directed graph produce a Morse decomposition for the vector field and furthermore, the MCG can be obtained from the tree that results from the collapsing each strongly connected component to a single vertex. Standard algorithms [32] indicate that this procedure can be performed in linear time in the number of vertices and edges in the graph. An implementation of Morse decomposition in practice can be found in [12].

A node in the MCG is an isolated invariant set, which may contain multiple fixed points and periodic orbits. For many engineering applications, such as the study of in-cylinder flow, engineers are often more concerned with individual fixed points and periodic orbits. Therefore, there is a need to build a graph $\mathscr{G}$, whose nodes consist of fixed points and periodic orbits. Similar to an MCG, the edges in $\mathscr{G}$ represents the connectivity information between the nodes according to the vector field. We refer to this graph as an *Entity Connection Graph*, or *ECG*. An ECG is a refinement of the MCG of the same vector field. In fact, an MCG can be obtained from the corresponding ECG by merging nodes that are in the same Morse set. Furthermore, the MCG is equal to the ECG when the vector field has a finite number of fixed points and periodic orbits, all of which have an isolating neighborhood of their own. Fig. 4 shows the difference between the MCG and ECG of a piecewise linear vector field created with our system. In the remainder of the paper, we will only show the ECG's for illustration purposes.

Given that the ECG is a refinement of the MCG, the reader may wonder why we emphasize the existence of both graphs. There are two reasons. The first is that we make use of information from the MCG to compute the ECG. The second has to do with the validity of the information. Any numerical or experimental method is subject to errors and thus one must be concerned with whether these errors are significant enough to produce misleading information. In the domain of numerical analysis the existence of spurious solutions would be an example of such misleading information. A rigorous analysis of the validity of the methods being presented here is beyond the scope of this paper, however we believe that as a basis for future research it is important to point out that the topological methods of Conley theory have been used to obtain computer assisted, but mathematically rigorous proofs concerning the structure of a wide variety nonlinear dynamical systems [33], [34]. Thus, our confidence level in the validity of the visualized structures and modifications is higher for those objects identified with the MCG than the ECG. Note that the ECG graph will not be complete without boundary analysis.

B. Vector Field Simplification on Surfaces

Vector field simplification corresponds to a reduction in the number of Morse sets in the decomposition (compare the two fields in Figure 3). Vector field modification corresponds to a change in the dynamics within an isolating neighborhood of a Morse set. To foreshadow the discussion of Section VI and to understand the potential vector field simplification that could possibly be associated with such a reduction requires the introduction of a topological invariant, the Conley index.

While the Conley index is applicable in the setting of a general dynamical system, we restrict our attention to the setting of flows on surfaces. An isolating neighborhood N is an *isolating block* if there exists $\varepsilon > 0$ such that for every $x \in \partial N$, we have

$$\varphi((-\varepsilon, 0), x) \cap N = \emptyset \quad or \quad \varphi((0, \varepsilon), x) \cap N = \emptyset$$

In other words, the trajectory enters N, leaves N, or both immediately everywhere on ∂N. The *exit set* of an isolating block N is $L := \{x \in \partial N \mid \varphi((0, \varepsilon), x) \cap N = \emptyset\}$. The pair (N, L) is called an *index pair*. In [11] it is proven that the sets in phase space which correspond to the strongly connected components are isolating blocks for the flow φ associated with the vector field V.

Let S be the maximal invariant set in the isolating block N with exit set L. The *Conley index* of S is the relative homology [35] of the index pair (N, L); that is, $CH_*(S) := H_*(N, L)$ (see Appendix for more details). Because we are restricting our attention to flows on orientable surfaces, it is sufficient to remark that we can write $CH_*(S) = (\beta_0, \beta_1, \beta_2) \in \mathbf{Z}^3$ where β_i represents the i-th Betti number of $H_*(N, L)$. It should be remarked that algorithms for computing Betti numbers exist [35] and thus we need not concern ourselves with these issues.

C. Important Conley Indices

Returning to the topic of design, the most important Conley indices are as follows:

$$\begin{aligned} x_0 \text{ an attracting fixed point} &\Rightarrow CH_*(x_0) = (1,0,0) \\ x_0 \text{ a saddle fixed point} &\Rightarrow CH_*(x_0) = (0,1,0) \\ x_0 \text{ a repelling fixed point} &\Rightarrow CH_*(x_0) = (0,0,1) \\ \Gamma \text{ an attracting periodic orbit} &\Rightarrow CH_*(\Gamma) = (1,1,0) \\ \Gamma \text{ a repelling periodic orbit} &\Rightarrow CH_*(\Gamma) = (0,1,1) \\ S = \emptyset &\Rightarrow CH_*(S) = (0,0,0) \end{aligned}$$

Observe that the emptyset is by definition an isolated invariant set. $(0,0,0)$ represents the index information for a region in which every point leaves in both forward and backward time. It should be noted that the reverse implications are not true. For example, given a polygonal index pair (N,L) for a vector field V, if $H_*(N,L) = (0,0,0)$, then one cannot conclude that the maximal invariant set in $\mathrm{cl}(N \setminus L)$ is the empty set. However, it can be proven that there does exist a different vector field $\bar{V}$ such that $V = \bar{V}$ on $\partial(\mathrm{cl}(N \setminus L))$ and the empty set is the maximal invariant set in $\mathrm{cl}(N \setminus L)$ under the flow induced by $\bar{V}$. Note the Poincaré index for an attracting fixed point is the same as a repelling one. Furthermore, the Poincaré index for a periodic orbit is zero, which equals that of an emptyset. Therefore, Poincaré index theory does not provide enough utility to handle periodic orbits, thus limiting its potential uses.

To make it clear how the Conley index information can be used in the vector field design process, let us review our strategy. The first step is the identification of a Morse decomposition for the entire flow. Given the associated MCG, the user identifies an interval that contains the elements which are to be eliminated. The interval defines an isolated invariant set for which an appropriate isolating block is constructed. The Conley index is then computed. This index information provides a topological constraint on the possible simplification or modification of the vector field within the isolating block. For example, if the Conley index does not equal $(0,0,0)$, then any modification will result in the existence of a nontrivial invariant set. To provide an even more specific example, if the Conley index is that of a fixed point, then any modification of the dynamics on the region will result in a vector field that possess at least one fixed point. Further examples will be provided in Section VI.

D. Vector Field Representation

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 only, and interpolation is used to obtain values on the edges and inside triangles. This applies to vector field editing, simplification, and analysis such as fixed point and periodic orbit extraction.

For the planar case, we use the popular piecewise linear interpolation method [1]. On curved surfaces, we borrow the interpolation scheme of Zhang et al. [6], 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.

E. Constrained Optimization

One of the essential operations in our system is constrained optimization, which refers to solving a vector-valued discrete Laplacian equation over a region N in the domain (a triangular mesh) where the vector values at the boundary vertices of N are the constraints. This operation is used to create periodic orbits (Section IV) and to perform topological simplification (Section VI). The equation has the following form:

$$\overline{V}(v_i) = \sum_{j \in J} \omega_{ij} \overline{V}(v_j) \tag{4}$$

where v_i is an interior vertex, v_j's are the adjacent vertices that are either in the interior or on the boundary of N, and V represents the vector field. The weights ω_{ij}'s are determined using Floater's mean-value coordinates [36]. Equation 4 is a sparse linear system, which we solve by using a conjugate gradient method [37]. For convenience, we refer to a vertex v as being *fixed* if the vector value at v is part of the constraints. Otherwise, v is *free*. Note that a similar formulation has been used to reduce the complexity of vector fields [6] and tensor fields [38].

IV. Periodic Orbit Creation

In this section, we describe novel algorithms for creating periodic orbits in the plane and on surfaces. The input to our algorithms consists of the desired type of the orbit (attracting or repelling) and a prescribed path, which is an oriented loop. Figure 5 shows an example path (left: blue loop). We then generate a sequence of evenly-spaced sample points on the loop (middle: green dots) and treat the tangent vectors at these points as constraints (middle: magenta arrows). Finally, we produce a vector field with a periodic orbit that closely matches the user input (right: red dashed lines). We use the dashed lines to represent the continuous periodic orbit so that it can be visually compared with the user-specified path. Next, we describe two ways of creating a vector field based on the constraints: basis vector fields and constrained optimization.

A. Attracting and Repelling Basis Vector Fields

An intuitive way to build a vector field that satisfies the constraints is to use basis vector fields [2] [17]. In this approach, every user-specified constraint is used to create a basis vector field defined in the plane. A vector field is then constructed as a weighted sum of these basis vector fields [17] [6]:

$$V(P) = \sum_i \omega_i(P) V_i(P) \tag{5}$$

where P is any position in the vector field, $V_i(P)$ is the i^{th} basis vector field, that refers to either a singular or a regular design element [6], and $\omega_i(P)$ is the weight for the i^{th} basis vector field. In our implementation, we use $\omega_i(P) = e^{\|P-P_i\|^2}$ for the i^{th} basis vector field, where P_i is the center position of the i^{th} basis vector field.

This idea has been applied to creating wind forces to guide computer animation [39], to testing a vector field visualization technique [17], and to generating vector fields for non-photorealistic rendering and texture synthesis [6].

In theory, any vector field can be created by using regular elements. In practice, however, it often requires an excessive number of regular elements to generate certain vector field

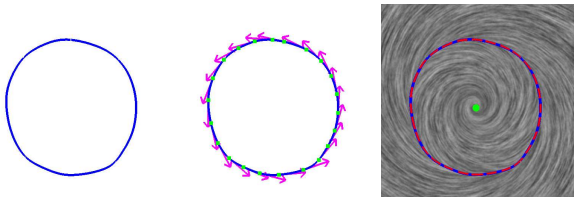


Fig. 5. Given an oriented loop (left), our system produces a sequence of sample points (middle: dots) and evaluates tangent vectors at those locations (middle: arrows). We then compute a vector field that contains a periodic orbit (right: red dashed lines) by generating constraints based on these vector values. Notice that the periodic orbit matches closely the user-specified loop.

features. For example, at least three regular elements are needed to specify a source or a center. To produce a periodic orbit, regular elements must be specified not only along the prescribed path, but also near the orbit in order to enforce the type of the orbit (attracting or repelling). Given that the cost of summing basis vector fields is proportional to the number of design elements, we wish to reduce the number of basis vector fields while maintaining efficient control. This is achieved with the introduction of two new types of design elements: *attachment elements* and *separation elements*.

Before describing these elements, we briefly review the concepts of attachment and separation points from Kenwright [40]. Given a vector field V and a point $\mathbf{p}_0$ in the plane, we consider the following two values: $e_1 \times u$ and $e_2 \times u$, where u is the vector value at $\mathbf{p}_0$ and e_1 and e_2 are the major and minor eigenvectors of the Jacobian. $\mathbf{p_0}$ is an *attachment point* if $e_1 \times u = 0$, and a *separation point* if $e_2 \times u = 0$. An attachment line consists of attachment points. Geometrically, such a line attract nearby flow. A separation line can be defined in a similar fashion except that nearby flow is repelled from the curve. Ideally, an attachment element will result in a basis vector field that has an attachment line as illustrated in Figure 6 (middle). The following formula describes an attachment element that has a desired vector value of $(1,0)$ at (x_0,y_0).

$$V(x,y) = B(x,y)\begin{pmatrix} 1 \\ c(y-y_0) \end{pmatrix} \quad (6)$$

where $B(x,y) = e^{-((x-x_0)^2+(y-y_0)^2)}$ is the blending function for the element and $c<0$ is a parameter that describes the speed at which the flow leaves the line $y = y_0$. The larger $|c|$ is, the more quickly the vectors near the attachment line point towards it. Notice the basis field contains an attachment line at $y = y_0$. Formula 6 can also be used to specify a separation element ($c>0$) and a regular element ($c=0$). When the vector value is $(\cos\theta_0, \sin\theta_0)$ for some constant θ_0, the formula has the following form:

$$V(x,y) = B(x,y)(\begin{pmatrix} \cos\theta_0 \\ \sin\theta_0 \end{pmatrix} + cP(x,y)\begin{pmatrix} -\sin\theta_0 \\ \cos\theta_0 \end{pmatrix}) \quad (7)$$

where $P(x,y) = -\sin\theta_0(x-x_0) + \cos\theta_0(y-y_0)$ is the signed distance of a point (x,y) to the line that is specified by the location and direction of the design element. Figure 6 compares two basis vector fields generated from a regular element (left) and an attachment element (middle). The right image shows an attracting periodic orbit created from four attachment elements. The ideas of attachment and separation will be used again in our periodic orbit extraction algorithm (Section V-A).

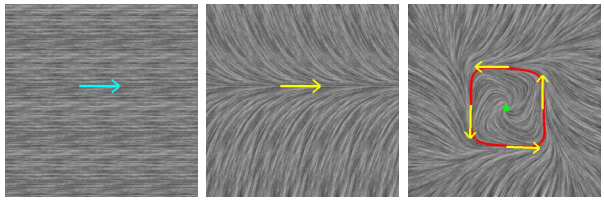


Fig. 6. This figure compares the basis vector field corresponding to a regular element (left) and an attachment element (middle). The periodic orbit in the right was created by using four attachment elements.

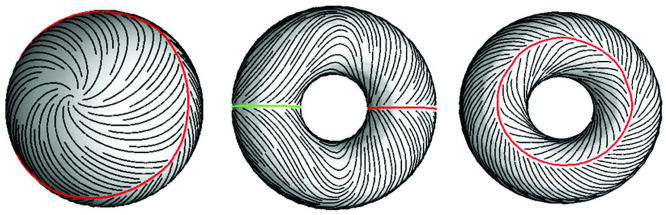


Fig. 7. Example vector fields created using our design system.

Vector field design using basis vector fields is intuitive and generates smooth results. However, the cost associated with this approach is proportional to the number of basis vector fields. To specify a relative large periodic orbit with high curvature often requires hundreds of attachment or separation elements, which makes interactive design a difficult task. The problem is magnified on surfaces on 3D as every basis vector field requires a global surface parameterization that is specific to the underlying design element [6]. Constructing hundreds of surface parameterizations makes it impractical to create a periodic orbit interactively. Next, we describe a different strategy that is based on constrained optimization.

B. Constrained Optimization for Periodic Orbit Creation

Given a user-specified oriented loop γ and the desired type of the periodic orbit, our system performs the following operations to create a periodic orbit closely matching the input.

First, we identify a region R_γ, which is a set of triangles that enclose γ. Next, we assign vector values to the vertices of R_γ according to the desired type, path, and orientation of the periodic orbit. Finally, our system performs a constrained optimization to compute vector values for vertices outside R_γ, i.e., the free vertices in the domain. The quality of the resulting periodic orbit depends on the choice of R_γ and the vector assignment on the boundary of R_γ.

We reuse attachment and separation elements to obtain vector values on R_γ. Basically, each line segment on the loop γ is used to infer a design element. We then compute vector values at the vertices of R_γ using the basis vector fields corresponding to these elements. Note when R_γ is chosen to be the whole domain, this technique becomes the basis vector field method mentioned earlier, which is computationally expensive. In practice, we choose R_γ be the smallest triangle strip containing γ. This greatly reduces the amount of computation that is associated with basis vector fields. In addition, it seems to produce reasonable results both on the plane and surfaces. We further speed up the process by only evaluating a basis field at the three vertices of the triangle that contains the corresponding element. When a vertex is shared by more than

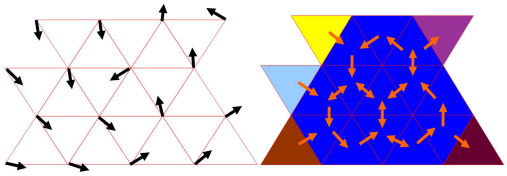


Fig. 8. An example for building a directed graph based on the input vector field defined on a triangular mesh. In the obtained directed graph, each node refers to a particular triangle, the direction of each directed edge are determined by the type of the edge. Based on the input vector field (left), we build the directed graph and compute the strongly connected components in the graph (right).

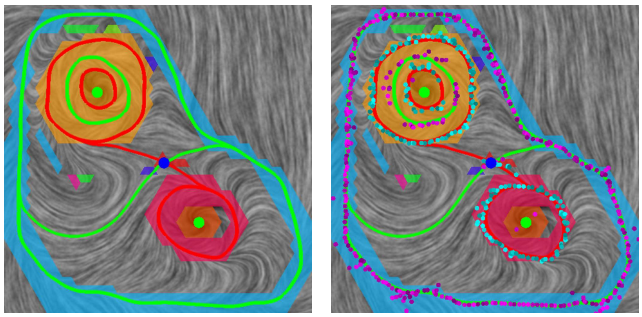


Fig. 9. An example of our periodic orbit detection algorithm. First, we compute strongly connected components and only consider components where periodic orbits may exist (left: colored regions). Next, we extract attachment points (right: cyan) and separation points (right: magenta) on the interior edges in these connected components. By combining the ideas of strongly connected components with the extraction of attachment and separation points, our algorithm is fast and efficient in finding periodic orbits.

one triangle in R_γ, we simply take the average of the vector values computed from each incident triangle. Fig 5 shows that this method tends to produce a periodic orbit (right: dashed red loop) that matches the user-specified loop (right: blue loop). To obtain smoother results, a larger R_γ can be constructed.

We have also extended a similar framework to create fixed points on surfaces. Every fixed point results in three constraints on the vertices that contains the desired fixed point. Vector values elsewhere in the mesh are obtained through constrained optimization. This framework avoids the need to construct a surface parameterization for each basis [6] and makes it possible to interactively create periodic orbits on surfaces in 3D. Figure 7 shows a number of vector fields that were created using our system. Although it works very well in practice, we should point out that our approach does not guarantee to create a periodic orbit according to user input.

V. Topological Analysis of Periodic Orbits

In this section, we describe a process in which an ECG is constructed and illustrated according to the Morse decomposition of a vector field (Section III). Because periodic orbits are essential features in a non-linear vector field, we need the ability to detect and locate periodic orbits in a fast and accurate manner. We will first present a new algorithm for periodic orbit identification before returning to ECG constructions.

A. Periodic Orbit Detection

Our periodic orbit detection method is inspired by Wischgoll and Scheuermann [25], in which they locate periodic orbits in a planar vector field by starting streamline tracing from a neighborhood of a fixed point and keeping track of repeated cell cycles. While this method is capable of detecting many periodic orbits, it assumes that any periodic orbit can be approached by a fixed point, which is not always true. One example case is the repelling periodic orbit between the two surrounding attracting orbits in Figure 9. To be able to detect periodic orbits even when they are not approached by any fixed point, we have developed a new periodic orbit detection method that has drawn ideas from the Morse decomposition [41] and separation and attachment lines [40].

A periodic orbit must situate inside a region of flow recurrence, which corresponds to certain types of strongly-connected components in the domain (Section III). Recall that strongly-connected components are computed by treating the mesh as a graph and merging triangles that share mixed edges. Figure 8 illustrates the construction of a directed graph from an input vector field defined on a mesh. Note that a strongly-connected component does not contain a periodic orbit if it either consists of a single triangle or it is a topological disk and contains no fixed point. This computation corresponds to computing a Morse decomposition of the flow. Figure 9 shows an example of the strongly connected components that may contain periodic orbits (left: colored regions).

Recall that multiple periodic orbits may exist in an isolated Morse set (a strongly-connected component). To extract individual periodic orbits in a fast and efficient manner, we need a good geometric indicator as to which strongly-connected components might contain periodic orbits. Kenwright presents efficient techniques in extraction open and closed separation and attachment lines [40]. We have used separation and attachment elements to create periodic orbits (Section IV-A). We now apply these ideas to periodic orbit extraction. Our algorithm is as follows:

1) *Step 1*: We compute the strongly connected components of the mesh according to the flow. In addition, we discard components that do not contain a periodic orbit, i.e., if the component S consists of a single triangle or if S is a topological disk and contains no fixed points. Let $\mathscr{S}$ be the set of strongly connected components that may contain a periodic orbit.
2) *Step 2*: We extract the attachment and separation points for every edge in the *interior* of a strongly connected component in $\mathscr{S}$.
3) *Step 3*: For every strongly connected component $S \in \mathscr{S}$, we start streamline tracing for each attachment point in S according to the flow. If the streamline reaches a fixed point or the boundary of S, we stop tracing and discard the attachment point. Otherwise, the streamline will approach an attracting periodic orbit. In case the periodic orbit has been discovered previously, it will be ignored. Otherwise, the periodic orbit is recorded, and a sequence of dense and evenly-spaced points are placed along the orbit. These points allow tracing from subsequent attachment points to quickly determine whether it is approaching an existing or new periodic orbit.
4) *Step 4*: We locate the repelling periodic orbits by repeating step 3 with the following two modifications: tracing will now (1) start from separation points, and (2) be in the backward direction of flow.

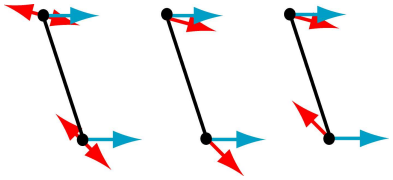


Fig. 10. An example scenario in which inconsistent tensor assignment can lead to false separation or attachment points. In the left image, given the vector values u at the vertices of an edge (cyan arrows) and the Jacobian tensor (red arrows represent the major eigenvectors e_1), it is clear that there is not any separation point on the edge. However, by converting the tensor field into a vector field (middle and right) and evaluating $e_1 \times u$ can cause false separation point to appear (right).

Kenwright evaluates $e_i \times u$ at the vertices of the edge and use linear interpolation to locate attachment and separation points. This formulation assumes that an eigenvector field can be treated as a vector field. However, as pointed out by Zhang et al. [42], treating an eigenvector field as a vector field will lead to discontinuities in the vector field and cause visual artifacts in tensor field visualization and non-photorealistic rendering. We have observed similar problems during the computation of attachment and separation points. For instance, consider the example shown in Figure 10, in which the vector field is constant along an edge e (cyan arrows) and the Jacobian along the edge is nearly constant (major eigenvectors are shown in red bidirectional arrows). When choosing a consistent direction assignment for the eigenvectors at the vertices (middle), we conclude that no separation or attachment point exists on e. However, the assignment in the right will lead to a false identification of a separation point. To overcome this problem, we simply assume the Jacobian is constant along an edge and evaluate it at the middle of an edge by performing linear interpolation on the Jacobians at the vertices. This efficiently removes the need to carefully assign directions to eigenvectors at the two vertices of an edge.

To perform tracing on surfaces, we use a Runge-Kutta scheme [43] that has been adapted to surfaces with a piecewise interpolation scheme that guarantees vector field continuity across vertices and edges [6].

We should point out that our approach does not guarantee to find all the periodic orbits in the given vector field.

B. ECG Construction and Display

Since the construction of the MCG graph follows from the work of Kalies et al. [11] and Kalies and Ban [12], we turn to a description of our algorithm for constructing the ECG for a given vector field starting with a brief review of vector field topology that involves fixed points and periodic orbits.

Consider a vector field V on a surface S that contains at least a fixed point or periodic orbit, i.e., the ECG of V is not empty. V induces a partition of S. Each sub-region in the partition is a *basin* that can be bounded by fixed points, periodic orbits, and/or separatrices. A streamline inside a basin flows from a source object α to a destination object ω. Both α and ω can be a node fixed point (a source or a sink) or a periodic orbit. In addition, for each of the three cases (node-node, node-periodic orbit, and periodic orbit-periodic orbit), the link between α and ω can be either direct, i.e., there is an edge connecting them in the ECG, or indirect, i.e., they are connected to some common saddles through separatrices. Note that a periodic orbit separates nearby flow into two parts. On either side, there can be one or more basins. When there is one basin, the periodic orbit is directly linked to a node or another periodic orbit. In the case of multiple basins, the periodic orbit is linked to other nodes or periodic orbits through saddles.

To compute the ECG, we perform a three-stage operation. First, we locate the fixed points and periodic orbits. These are the nodes in the ECG. Next, we compute all the separatrices by tracing from every saddle in its incoming and outgoing directions until the trajectories end in a node or a periodic orbit. Finally, we identify edges in the ECG that are not separatrices. Our methods for fixed point extraction and separatrix computation are according to Helmann and Hesselink [22] except they do not handle vector fields that contain periodic orbits. Periodic orbits are identified using the algorithm described in Section V-A.

We now describe how to compute non-separatrix edges in the ECG. As discussed earlier, this corresponds to an edge in the ECG that does not involves any saddle. There are four cases: (1) a source and a sink (type 1), (2) a source and an attracting periodic orbit (type 2), (3) a sink and a repelling periodic orbit (type 3), and (4) a repelling periodic orbit and an attracting periodic orbit (type 4). Note a node can only be involved in one non-separatrix edge, and so does each side of a periodic orbit. We use a flag to describe every node. The flag is set to 1 if the node is connected to a saddle in the ECG. Otherwise, the flag is set to 0. Similarly, we define a flag for each side of a periodic orbit to record whether there is at least one separatrix approaching the periodic orbit from that side. To compute non-separatrix edges, we first locate edges emanating from repelling orbits. For each repelling periodic orbit γ and each side, if the corresponding flag is 0, we find a nearby point on that side of γ and perform tracing in the direction of the flow until the streamline terminates at a sink or an attracting periodic orbit. In case of a sink, we mark its flag to be 1 and insert an edge (type 3) in the ECG. If the streamline ends in an attracting periodic orbit, we mark the flag to be 1 for the side of the attracting orbit from which the streamline approaches. An edge (type 4) is then inserted into the ECG. Notice that at the end of this step, all non-separatrix edges of types 3 and 4 are found. We now perform the same operations to all the extracted attracting periodic orbits whose side or sides are still marked as 0, except that tracing is now done in the reverse direction of the flow. This allows us to find all type 2 edges. Finally, we go through every source that still has a flag of 0 and trace from a nearby point in the forward direction until it terminates at a sink. This will find all the type 1 edges. It appears that type 1 edges are rather uncommon. In fact, the only instance that we know of is the idealized magnetic field over a sphere, which contains two fixed points and no periodic orbits. Figure 11 illustrates this process with an example vector field that contains two sources, one saddle, and four periodic orbits. In (a), we extract the fixed points and periodic orbits. We also mark as unvisited (white disks) for all the sources and sinks and for both sides of every periodic orbit. Next (b), we compute separatrices and mark as visited (black disks) any node or any side of a periodic orbit that is

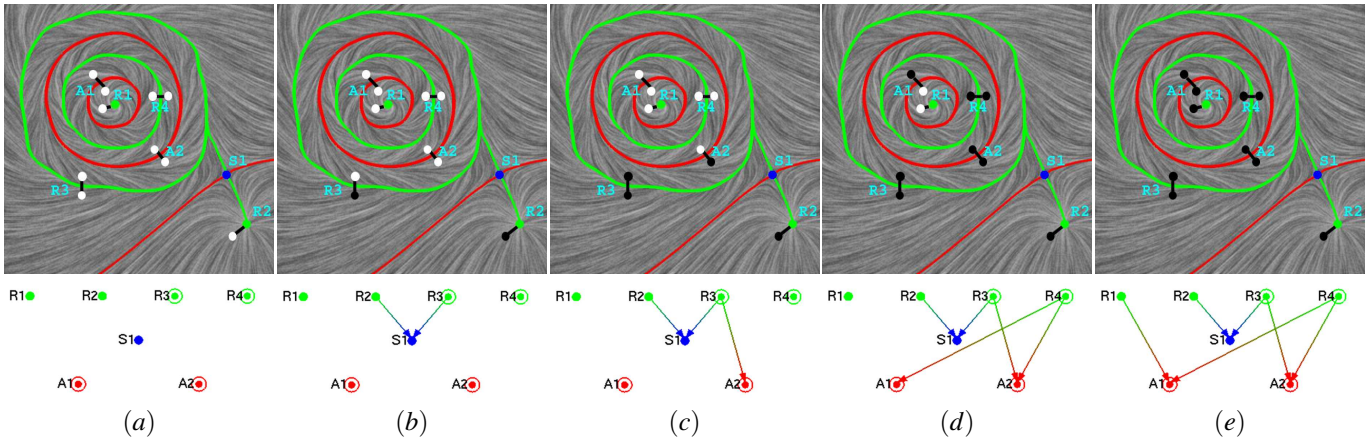


Fig. 11. This figure illustrates our algorithm for construction of ECG's. First (a), we perform fixed point and periodic orbit extraction. We mark as unvisited (white disks) for every source/sink and for both sides of every periodic orbit. Next (b), we compute all the separatrices and mark as visited (black disks) for $R2$ and the outer side of $R3$ since they are connected to the saddle $S1$ in the ECG. In (c), we start from the inner side of $R3$ and follow the flow forward to find the link to the outer sider of $A2$. An edge is added to the ECG, and both sides in the link are now marked as visited. In (d), we perform similiar operations to the unvisited sides of every repelling orbits (both sides of $R4$) to find all the links to a sink or an attracting orbit. Finally (e), we start from any unvisited side of an attracting periodic orbit and follow the flow in the reverse direction to locate links to unmarked sources.

connected to a saddle. In the next stage, we start from any unvisited side of a repelling periodic orbit and follow the flow forward to locate links to a sink or an attracting orbit. In (c), such an operation found a link between the inner side of $R3$ and the outer side of $A2$, both of which are now marked as visited. Performing this operation on all the extracted repelling periodic orbits leads to (d), in which links such as $R4$/$A1$ and $R4$/$A2$ are found. Finally (e), we start from any unvisited side of an attracting periodic orbit and follow the flow in the reverse direction to locate the remaining edges in the ECG.

To display an ECG, we arrange vector field features (fixed points and periodic orbits) in three rows, with sources and repelling period orbits in the top row, sinks and attracting period orbits in the bottom row, and saddles in the middle row (Figure 3). We also provide the user with the capability to select an object either in the flow display or the graph display, and our system will highlight the object in both screens. This allows a user to navigate through a rather complex flow field with relative ease.

C. Application to Analytic Data

For all the fields designed with our system, we use this method to detect periodic orbits and construct ECG's. In addition, we have tested our method on other datasets generated from mathematical formulas and from fluid simulation. Figure 12 shows a vector field that corresponds to

$$V(x,y) = \begin{pmatrix} y \\ -x + y\cos(x) \end{pmatrix} \tag{8}$$

It has been proven that this system has exactly n periodic orbits in the region $\sqrt{x^2+y^2} < (n+1)\pi$ [44]. We sample the vector field at the vertices of a bounded underlying mesh, and employ the piecewise linear interpolation scheme [1] to obtain values inside triangles. The left of this figure shows the periodic orbits extracted using our method, and the right portion displays the corresponding ECG. There are five periodic orbits. Notice our method is able to detect periodic orbits even when there are no saddles in the field.

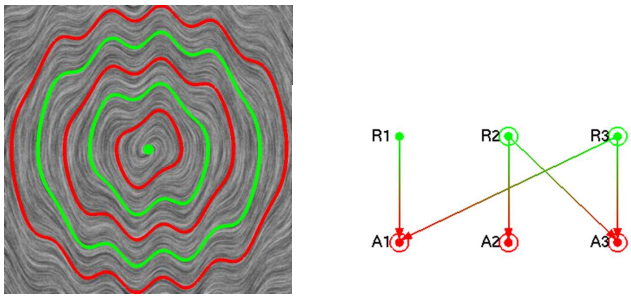


Fig. 12. The vector field defined in Equation 8 over the region $\{(x,y)|max(|x|,|y|) < 11\pi\}$. There is one source in the region enclosed by five periodic orbits. Our algorithm was able to capture all of these orbits without requiring the presence of any separatrices.

D. Application to Engine Simulation Data

We have also applied our technique to two datasets from automotive engine simulation [7], more specifically, the design and optimization of in-cylinder flow. Engineers responsible for the design of, in this case, a diesel engine try to create an ideal pattern of motion, which can be described by a swirling flow around an imaginary axis. Achieving these ideal patterns of flow optimizes the mixture of oxygen and fuel during the ignition phase of the valve cycle. Optimal ignition leads to very desirable consequences associated with the combustion process including: more burnt fuel (less wasted fuel), lower emissions, and more output power. One type of flow, referred to as the *swirl motion*, is shown in Figure 13 (right). Such an ideal is often strived for diesel engines.

In Figure 1 we visualize the flow and its topology inside the combustion chamber from the diesel engine simulation. We have sliced through the geometry in the same manner that engineers do when analyzing the simulation results. The first slice, at 10% the length of the volume, indicates a swirl pattern that deviates rather strongly from the ideal -which would result in a simple recirculation orbit around the center. The second slice, at 25% down the chamber geometry we see a periodic orbit very close to the center that starts to approximate the ideal swirl motion. However, other less ideal fixed points are found near the perimeter of the geometry. The method we use

Fig. 13. Idealized in-cylinder flow through a gas engine (left) and a diesel engine (right). Figures 14 and 2 show our visualization of CFD data simulating such flows.

here is similar to the moving cutting plane topology approach of Tricoche et al. [45]. We note that caution must be used when interpreting these results since the vector field has been projected onto 2D slices. On the other hand, the engineers involved are very familiar with the simulation data and are well aware of its overall characteristics.

Figure 2 shows from two viewpoints some simulation result in which undesired fixed points and periodic orbits are present. There are a total of 226 fixed points and 52 periodic orbits. The total time to construct the ECG for the flow is 29.15 seconds on a 3.6 GHz PC with 3.0 GB RAM. Another type of motion, termed *tumble flow*, is shown in Figure 13 (left). The axis of rotation in the tumble case is orthogonal to that of the swirl case. The dataset that is being visualized (Figure 14) is also from simulation, and it contains 56 fixed points and 9 periodic orbits. The ECG for this dataset is shown in the bottom row. Through the application of our automatic periodic orbit extraction and visualization algorithm we can observe a closed streamline about a central axis corresponding to the ideal pattern of tumble motion in the gas engine simulation results. This is precisely the type of re-circulation that the engineers strive to realize when designing the intake ports of a gas engine cylinder. Our algorithm enables the CFD engineers to automatically detect and visualize this highly sought-after pattern of flow in a direct manner for the first time (see Figure 14). The total time for computing the ECG of this time is 31.58 seconds. The ECG produced from the diesel engine simulation results is of even higher complexity than that of the gas engine. Table I shows the complexity for both simulation datasets and the timing results in seconds.

VI. Vector Field Simplification

Topological simplification of a vector field has many applications, such as flow visualization, texture synthesis, and non-photorealistic rendering.

A. Single Attractor/Repeller Pair Cancellation

A well-known topological simplification operation is *pair cancellation* on a pair of fixed points with opposite Poincaré indices and a unique orbit connecting them. This operation has also been referred to as pair annihilation [1]. After cancellation, both fixed points disappear. Tricoche et al. [1] perform this operation in planar domains based on Poincaré index theory, which does not apply to periodic orbits. Zhang et al. [6] provide an efficient implementation of the pair

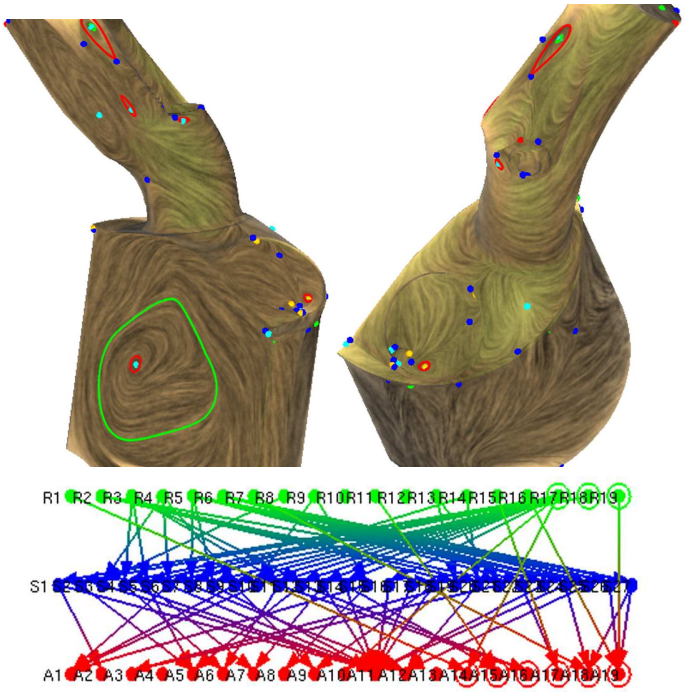


Fig. 14. The visualization of CFD data simulating in-cylinder flow through a gas engine from two viewpoints (top), and the corresponding ECG (bottom). Through the application of our automatic periodic orbit extraction algorithm we can observe a closed streamline about a central axis corresponding to the ideal pattern of tumble motion in the gas engine simulation results. This is precisely the type of re-circulation that the engineers strive to realize when designing the intake ports of a gas engine cylinder (Figure 13, right).

cancellation operation based on Conley index theory. They also extend fixed point pair cancellation to surfaces and for pairs that are not connected by a separatrix, such as a center and saddle pair. However, neither technique deals with periodic orbits, which limits their potential applications in visualization and graphics. Our paper addresses this by providing a general framework that allows cancellations of a repeller and attractor pair in which either object or both can be a periodic orbit. Similar to Zhang [6], our framework is based on Conley index theory. Before providing the details on our general framework, we first comment on what we mean by pair cancellation.

Pair cancellation P involves a repeller R and an attractor A. P is *direct* if there is at least one edge between R and A in the ECG, and P is *indirect* if R and A are linked through either one or two saddles. When a node or a periodic orbit is linked to a saddle through one connecting separatrix, the pair are *singly connected*. Otherwise, they are *doubly connected*. We have identified six direct cancellation scenarios (Figure 15) and seven indirect ones (Figure 16) on the plane. Our system can handle all of these cases. To our best knowledge, previous pair cancellation methods are only available to handle case (1) in Figure 15.

When performing pair cancellation, we expect the complexity of the flow to be reduced near the object pair, such as the case in Figure 15 (1). However, the reduction in the complexity does not mean the resulting flow will always be free of fixed points and periodic orbits. For instance, a sink and periodic orbit cancellation results a source as shown in Figure 15 (2). In fact, the characteristic of the resulting flow is constrained topologically by the Conley index of the isolating block over

TABLE I

THE COMPLEXITY AND TIMING RESULTS FOR TWO CFD DATA SIMULATING IN-CYLINDER FLOW THROUGH A COMBUSTION ENGINE (FIGURES 14 AND 2). AN EDGE IN THE ECG CORRESPONDS TO A LINK BETWEEN A SOURCE AND DESTINATION OBJECT PAIR, IN WHICH BOTH OBJECTS CAN BE A FIXED POINT OR A PERIODIC ORBIT. TIMES (IN SECONDS) ARE MEASURED ON A 3.6 GHZ PC WITH 3GB RAM.

dataset name	# polygons	# fixed points	# periodic orbits	# edges in ECG	time extracting fixed points	time extracting periodic orbits	time computing edges	time total
gas engine	105,192	56	9	97	0.16	22.33	9.09	31.58
diesel engine	886,296	226	52	295	3.16	21.52	4.48	29.15

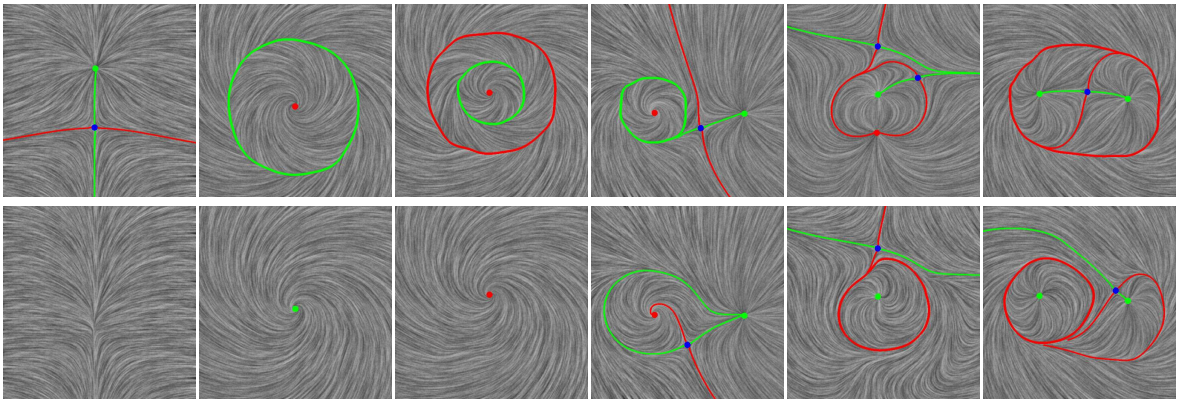


Fig. 15. The six direct cancelation scenarios: (1) a source and saddle with a unique connecting separatrix, (2) a sink and a periodic orbit, (3) an attracting periodic orbit and a repelling one, (4) a periodic orbit and a saddle with a unique connecting separatrix, (5) a sink and a saddle with two connecting separatrices, and (6) a periodic orbit and a saddle with two connecting separatrices. The top row shows the original vector fields, while the bottom row displays the vector field after cancellation. Notice that our cancellation operations are only applied to the intended objects.

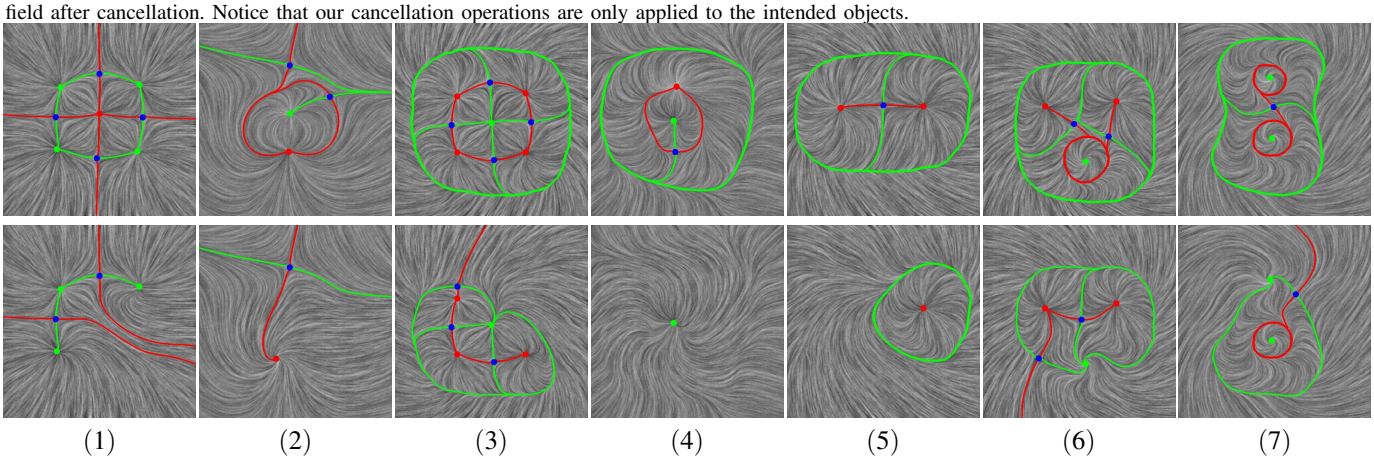


Fig. 16. The seven indirect cancellation scenarios: (1) a source and a sink pair with two saddles between them, (2) a source and a sink with one saddle between them, (3) a sink and a periodic orbit with two saddles between them, (4) a sink and periodic orbit with one saddle between them and two orbits between the saddle and the sink, (5) a sink and periodic orbit with one saddle between them and two orbits between the saddle and the periodic orbit, (6) two periodic orbits with two saddles between them, and (7) two periodic orbits with a saddle between them. The top row shows the original vector fields, while the bottom row displays the vector field after cancellation. Notice that our cancellation operations are only applied to the intended objects.

which the flow is modified. When cancelling a node and saddle pair, the Conley index of such a block is $(0,0,0)$, which is the same as a fixed point-free vector field. For a sink and periodic orbit pair, the Conley index is $(0,0,1)$ which is that of a source. Furthermore, pair cancellation does not always lead to simpler behaviors, such as Figure 15 (5). Cancelling a doubly-connected node-saddle pair leads to a periodic orbit. In fact, the only other case in which the flow is not simplified through pair cancellation is shown in Figure 15 (6), where a doubly-connected periodic orbit and saddle pair is replaced by another such pair. Both cases are direct cancellations of doubly-connect object pair. In all other cases, pair cancellation leads to simpler but not necessarily trivial flow.

We now describe our framework for a single pair cancellation that can now handle (1) periodic orbits, (2) doubly connections, and (3) indirect cancellation. Given a repeller R and an attractor A, our algorithm first searches the ECG to find the smallest interval that contains both R and A. This is achieved by finding all the nodes in the ECG that can both reach A and be reached from R. There are three possibilities: (1) R and A are directly related, (2) R and A are indirectly linked through a set of saddles S_i's, and (3) R and A are not connected. Case (3) will be ignored. Note that the first stage is conducted purely on the graph level. Let $\mathscr{R} = \{R\} \bigcup \{S_i\}$ and $\mathscr{A} = \{A\} \bigcup \{S_i\}$. Note when R and A are directly connected, the set of $\{S_i\}$ is empty. It should also be noted that Kalies

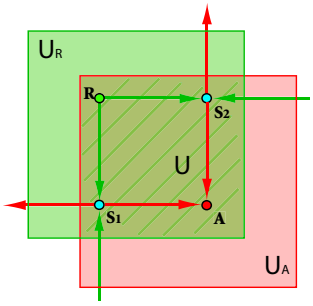


Fig. 17. An example shows the regions obtained for cancelling a repeller R and an attractor A pair. Our region growing first gets the region by following the flow forward from R and the interval S_i respectively to get U_R, then follows the flow backward from A and the interval S_i respectively to get U_A. $U = U_R \bigcap U_A$ is the region (the shadow region) in which we will perform smoothing.

and Ban [12] provide a dimension independent algorithm for determining intervals in a Morse decomposition.

In the second stage, we consider the minimal set of triangles in the domain that contain $\mathscr{R}$. We then grow from these triangles by adding one triangle at a time across mixed or exit edges. We now have a region U_R that contains all the triangles reachable from any object in $\mathscr{R}$. Then, we perform region growing from the minimal set of triangles that contain $\mathscr{A}$ by adding triangles across mixed or entrance edges. This results in a region U_A that consists of triangles that can reach any object in $\mathscr{A}$. $U = U_R \bigcap U_A$ is an isolating block that is necessary to perform pair cancellation.

In the last step, we replace the flow inside U by performing constrained optimization (Section III-E). While this method does not guarantee that the flow will be simpler, in practice we have observed that it performs well. Note that other methods can also be used to modify the flow.

This framework is illustrated as **Algorithm 1** with the following pseudocode.

Algorithm 1: A general framework for pair cancellation
Input: A vector field V, its ECG, a repeller R and an attractor A
Output: The vector field $\overline{V}$ after cancellation

Search ECG for any intermediate nodes S_i between R and A.
T_+ = set of triangles containing either R or S_i for some i.
Perform region growing from T_+ according to V by adding triangles across exit or mixed edges. Let U_R be the resulting set of triangles.
T_- = the set of triangles containing either A or S_i for some i.
Perform region growing from T_- according to $-V$ by adding triangles across entrance or mixed edges. Let U_A be the resulting set of triangles.
$U = U_R \bigcap U_A$(See Figure 17)
Perform vector field smoothing on the interior vertices of U accroding to Equation(4). The resulting vector field is $\overline{V}$.
Return $\overline{V}$.

For any pair cancellation operation relying on the ECG, it is possible that region growing from the repellers and attractors can "walk" over fixed points, periodic orbits, and separatrices that are not intended for cancellation. Including these triangles in the constrained optimization may cause unwanted topological modifications (*note this will not happen if one uses MCG to determine the cancellation operation*). To address this issue, we tag all the triangles in the mesh that contain either a fixed point, or part of a periodic orbit or separatrix. During the construction of isolating blocks, we do not allow triangles to be added if they are tagged and contain features not intended for cancellation.

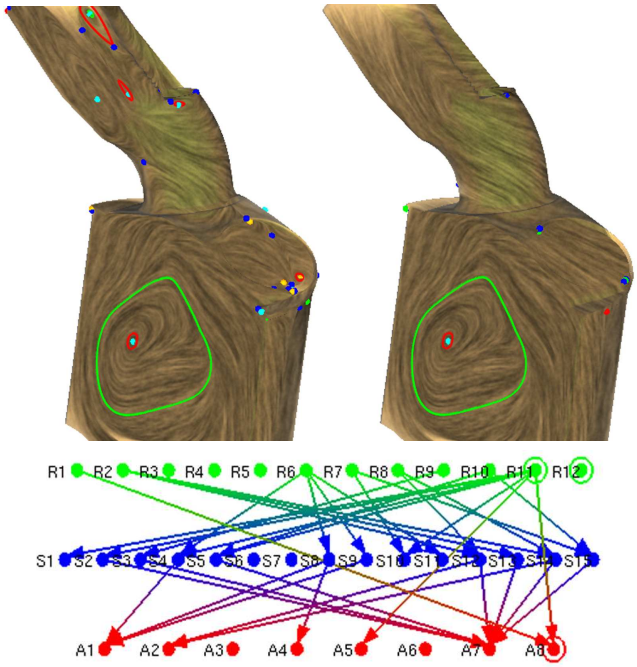


Fig. 18. User-guided flow smoothing on CFD data simulating in-cylinder flow through a gas engine: before (upper-left) and after (upper-right). Compare the ECG after smoothing (lower) with before smoothing (Figure 14, lower).

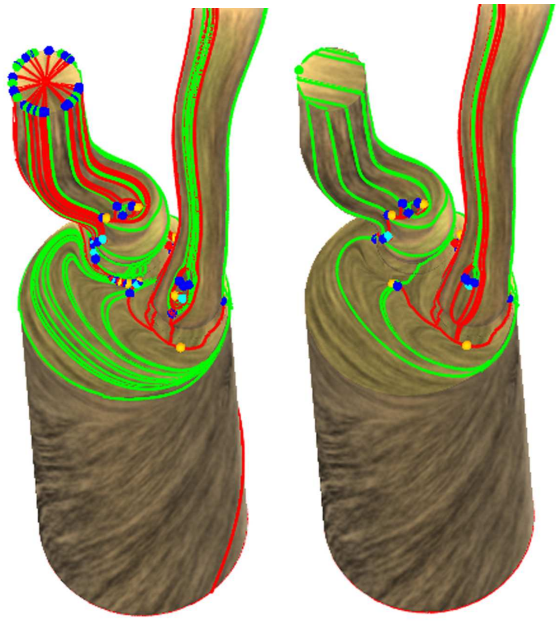


Fig. 19. User-guided flow smoothing on CFD data simulating in-cylinder flow through a diesel engine: before (left) and after (right).

B. User Guided Flow Smoothing

In the proceeding section, we have described techniques that automatically determine a region where the flow needs to be modified. Sometimes it is desirable to provide a user with control over the location and shape of the region. Zhang et al. [6] describe such an operation for graphics applications such as non-photorealistic rendering and texture synthesis. We apply their algorithm to large scale CFD simulation datasets. In addition, unlike Zhang et al. who accept a topological disk, we now allow a region to have any number of boundaries. Figure 18 shows the results of user-guided flow smoothing on

CFD simulation data of in-cylinder flow in a gas engine. The field on the upper-right was obtained by a sequence of five user-guided smoothing operations (the actual region boundaries are not shown). Notice the field is considerably simpler than the original field (upper-left). The simplified vector field retains the important larger scale tumble motion characteristics while smoothing non-ideal behavior. Also compare the ECG of the smoothed field (Figure 18, lower) with that of the field before smoothing (Figure 14, lower). Figure 19 compares the diesel engine dataset (left) with the one obtained from a series of six user-guided simplification operations (right). Flow smoothing is an efficient method of reducing the complexity of a vector field.

VII. Topology-Based Streamline Visualization

Visualization is crucial for the analysis and design of vector fields. Most existing visualization techniques, such as texture- and streamlines-based methods, are designed for fixed points. While they perform well for illustrating local patterns such as fixed points, other features (separatrices and periodic orbits) are often not well-preserved. In Figure 20, a vector field with three periodic orbits is depicted using IBFV [17] (a), and evenly-placed streamlines [46] (b). Notice that it is difficult to see periodic orbits and separatrices using texture-based methods such as IBFV. Streamline-based methods can better illustrate trajectories. However, most existing methods such as Jobard and Lefer [46] and Verma et al. [47] do not take into account periodic orbits or separatrices in seed placement and streamline termination criteria. This causes visual discontinuity in periodic orbits and missing separatrices.

Several researchers have incorporated vector field topology into texture-based methods [48]. Most of the figures in this paper are created in that fashion. On the other hand, streamline-based methods can better illustrate individual streamlines, which makes it an attractive approach when interactive display is not required. In this section, we describe a method for which vector field topology is used for streamline placement.

We adapt the evenly-placed streamline method of Jobard and Lefer [46] with the following modifications. First, we extract periodic orbits and separatrices in the vector field and make them the initial streamlines. To avoid visual clusterings near sources, sinks, and periodic orbits, we terminate a separatrix if it is within a distance from the non-saddle end. Next, we add additional streamlines in the same manner as Jobard and Lefer [46]. This modification ensures that vector field topology is maintained in the visualization and no visual discontinuity for periodic orbits (Figure 20, c). Finally, we highlight vector field topology with colors (d) such that attracting periodic orbits and outgoing separatrices from saddles are colored in red while repelling periodic orbits and incoming separatrices are colored in green. To avoid confusions near sources and sinks, the only fixed points we include in the visualization are saddles, which are colored in blue. Figure 7 shows additional examples. Notice a periodic orbit on a 3D surface (middle-left and middle right) is often partially visible from any given viewpoint. They are difficult to discern without being highlighted.

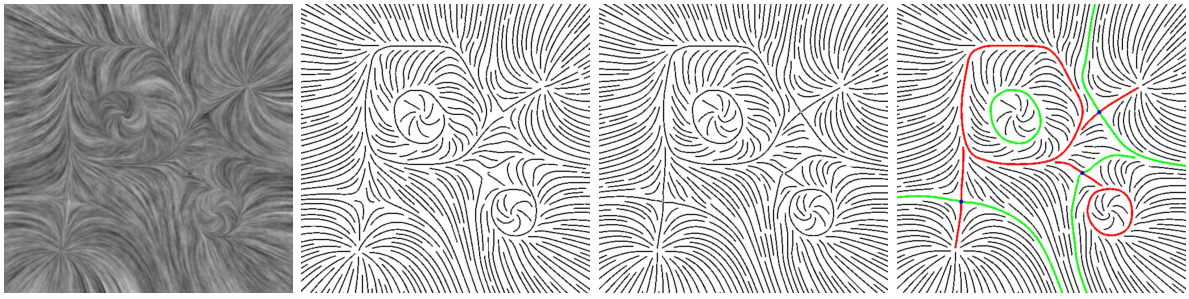
VIII. Conclusion and Future Work

In this paper, we have described a vector field design system in which fixed points and periodic orbits can be created, modified, and removed. At the core of our implementation are results from Conley theory, which enable a unified framework for the efficient control of fixed points and periodic orbits through editing operations such as feature cancellation. To our knowledge, this is the first vector field design system that addresses periodic orbits. As part of the system, we provide a novel technique for periodic orbit extraction by computing the strongly-connected components of the underlying mesh according to the flow and by extracting separation and attachment points. Furthermore, we define a new graph-based topological representation of a vector field, the ECG, and propose efficient techniques to construct the ECG. We have applied our vector field analysis and simplification techniques to an engineering application: visualizing flows from engine simulation. Both of our analysis and simplification techniques can handle vector fields on curved surfaces. Finally, we augment streamline-based vector field visualization techniques by including vector field topology (separatrices and periodic orbits) in the streamline generation and by highlighting them.

There are a number of future directions. First, our periodic orbit detection method depends on efficient extraction of separation and attachment points. While we have observed in our experiments that these points tend to be close to periodic orbits, a rigorous mathematical study on the subject is needed. Furthermore, other methods for extracting separation and attachment points, such as that of Peikert and Roth [49], may lead to more numerically stable results. Second, our current MCG and ECG construction methods assume closed surfaces. We are investigating means to extend them to handle surfaces with boundaries. Third, we plan to investigate automatic techniques for vector field simplification. Fourth, we are exploring more intuitive illustration of the ECG's. In particular, we plan to explore graph and network visualization techniques developed by researchers in the Information Visualization community. Fifth, more rigorous mathmatical analysis and proofs need to be introduced to guarantee the minimal structure will be obtained after vector field simplification. Finally, we plan to study the reconstruction of original vector field from the obtained ECG/MCG graph, which can be applied to vector field compression.

Acknowledgment

We would like to thank Greg Turk for his valuable ideas. We also appreciate the discussions with Andrzej Szymczak, Jarek Rossignac and John C. Hart. We are grateful to Juergen Schneider for his valuable past contributions. Finally, we wish to thank our anonymous reviewers for their valuable comments and suggestions. This work was funded by NSF grants CCF-0546881. Konstantin Mischaikow was partially supported by NSF grants DMS-0511115 and DMS 0443827, by DARPA, and by the U.S. Department of Energy. Paweł Pilarczyk was partially supported by the DARPA TDA project, as well as JSPS Postdoctoral Fellowship No. P06039 and Grant-in-Aid for Scientific Research (No. 1434055), Ministry of Education, Science, Technology, Culture and Sports, Japan.



(a) (b) (c) (d)

Fig. 20. An example of our streamline-based visualization technique on the plane: (a) a texture-based method (IBFV [17]), (b) a streamline-method [46], (c) our streamline method which uses vector field topology, (d) same image from (c) with periodic orbits and separatrices being highlighted. Notice with our method (c and d), vector field topology is well-maintained by streamlines and they are easily discernable.

APPENDIX
COMPUTATION OF CONLEY INDEX

The triangular mesh, which is the underlying domain for our system, consists of a collection of vertices, $\mathscr{V}$ whose elements are denoted by $\langle v_i \rangle$, edges $\mathscr{E}$, whose elements are denoted by $\langle v_i, v_j \rangle$, $i \neq j$, and triangles $\mathscr{T}$, whose elements are denoted by $\langle v_i, v_j, v_k \rangle$, $i \neq j \neq k \neq i$. Given a subset X of the mesh, let $\mathscr{V}(X)$, $\mathscr{E}(X)$, and $\mathscr{T}(X)$ denote the collections of vertices, edges, and triangles which are contained in X.

Let $C_0(X)$, $C_1(X)$, and $C_2(X)$ be the *free abelian groups on the sets* $\mathscr{V}(X)$, $\mathscr{E}(X)$, and $\mathscr{T}(X)$, respectively [50] and set $C_{-1}(X) = 0$. Recall, that if (N,L) is an index pair then $L \subset \partial N$. Since, N is given as the union of a set of triangles, $\mathscr{V}(L) \subset \mathscr{V}(N)$, $\mathscr{E}(L) \subset \mathscr{E}(N)$, and $\mathscr{T}(L) = \emptyset$. Thus, the quotient groups

$$C_i(N,L) := C_i(N)/C_i(L), \quad i = -1,0,1,2$$

are free abelian groups.

Since $\mathscr{V}(N)$, $\mathscr{E}(N)$, and $\mathscr{T}(N)$ induce a basis for $C_0(N)$, $C_1(N)$, and $C_2(N)$, respectively, to define group homomorphisms $\partial_i : C_i(N) \to C_{i-1}(N)$, for $i = 0,1,2$, it suffices to prescribe the action of ∂_i on the individual vertices, edges, and triangles. Define

$$\partial_0 \langle v_0 \rangle := 0 \tag{9}$$

$$\partial_1 \langle v_0, v_1 \rangle := \langle v_1 \rangle - \langle v_0 \rangle \tag{10}$$

$$\partial_2 \langle v_0, v_1, v_2 \rangle := \langle v_0, v_1 \rangle - \langle v_0, v_2 \rangle + \langle v_1, v_2 \rangle \tag{11}$$

The ∂_i are called *boundary operators*. Observe that they induce boundary operators on the quotient groups, $\partial_i : C_i(N,L) \to C_{i-1}(N,L)$.

Let $Z_i(N,L) := \{x \in C_i(N,L) \mid \partial_i x = 0\}$ and $B_i(N,L) := \partial_{i+1}(C_{i+1}(N,L))$. It is straightforward to check that $\partial_{i-1} \circ \partial_i = 0$, hence $B_i(N,L) \subset Z_i(N,L)$. The *relative homology* of the pair (N,L) are the quotient groups

$$H_i(N,L) := Z_i(N,L)/B_i(N,L) \quad i = 0,1,2.$$

Because we assume that the original triangular mesh is a closed orientable surface embedded in $\mathbf{R}^3$, the relative homology groups are free groups, that is

$$H_i(N,L) = \mathbf{Z}_i^{\beta}$$

where $\mathbf{Z}$ are the integers and β_i is a non-negative integer. β_i is called the i-th *Betti number*. Please refer to [35] for more details about Betti numbers and homology.

REFERENCES

[1] X. Tricoche, G. Scheuermann, and H. Hagen, "Continuous topology simplification of planar vector fields," in *Proceedings IEEE Visualization 01*, 2001, pp. 159–166.

[2] E. Praun, A. Finkelstein, and H. Hoppe, "Lapped textures," in *Proceedings of ACM SIGGRAPH 00*, 2000, pp. 465–470.

[3] G. Turk, "Texture synthesis on surfaces," in *Proceedings of ACM SIGGRAPH 01*, 2001, pp. 347–354.

[4] L.-Y. Wei and M. Levoy, "Texture synthesis over arbitrary manifold surfaces," in *Proceedings of ACM SIGGRAPH 01*, 2001, pp. 355–360.

[5] J. Stam, "Flows on surfaces of arbitrary topology," in *ACM Transactions on Graphics (SIGGRAPH 03)*, vol. 22, no. 3, July 2003, pp. 724–731.

[6] E. Zhang, K. Mischaikow, and G. Turk, "Vector field design on surfaces," *ACM Transactions on Graphics*, vol. 25, no. 4, pp. 1294–1326, 2006.

[7] 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 04*, 2004, pp. 51–58.

[8] J. Hale and H. Kocak, *Dynamics and Bifurcations*. New York: Springer-Verlag, 1991.

[9] M. W. Hirsch, S. Smale, and R. L. Devaney, *Differential equations, dynamical systems, and an introduction to chaos*, 2nd ed., ser. Pure and Applied Mathematics (Amsterdam). Elsevier/Academic Press, Amsterdam, 2004, vol. 60.

[10] C. Conley, *Isolated Invariant Sets and the Morse Index*. Providence, RI: AMS, 1978, cBMS **38**.

[11] W. D. Kalies, K. Mischaikow, and R. C. A. M. VanderVorst, "An algorithmic approach to chain recurrence," *Found. Comput. Math.*, vol. 5, no. 4, pp. 409–449, 2005.

[12] W. Kalies and H. Ban, "A computational approach to conley's decomposition theorem," *Journal of Computational and Nonlinear Dynamics*, vol. 1, no. 4, pp. 312–319, 2006.

[13] R. S. Laramee, H. Hauser, H. Doleisch, F. H. Post, B. Vrolijk, and D. Weiskopf, "The State of the Art in Flow Visualization: Dense and Texture-Based Techniques," *Computer Graphics Forum*, vol. 23, no. 2, pp. 203–221, June 2004.

[14] F. H. Post, B. Vrolijk, H. Hauser, R. S. Laramee, and H. Doleisch, "The State of the Art in Flow Visualization: Feature Extraction and Tracking," *Computer Graphics Forum*, vol. 22, no. 4, pp. 775–792, Dec. 2003.

[15] D. Weiskopf and G. Erlebacher, "Overview of flow visualization," in *The Visualization Handbook*. In C.D. Hansen, C.R. Johnson (Eds.), Elsevier, Amsterdam, 2005, pp. 261–278.

[16] A. Rockwood and S. Bunderwala, "A toy vector field based on geometric algebra," in *Proceeding Application of Geometric Algebra in Computer Science and Engineering, (AGACSE2001)*, 2001, pp. 179–185.

[17] J. J. van Wijk, "Image based flow visualization," in *ACM Transactions on Graphics (SIGGRAPH 02)*, vol. 21, no. 3, Jul 2002, pp. 745–754.

[18] J. van Wijk, "Image based flow visualization for curved surfaces," in *Proceedings IEEE Visualization 03*, 2003, pp. 123–130.

[19] R. S. Laramee, B. Jobard, and H. Hauser, "Image space based visualization of unsteady flow on surfaces," in *Proceedings IEEE Visualization 03*, October 2003, pp. 131–138.

[20] H. Theisel, "Designing 2d vector fields of arbitrary topology," in *Computer Graphics Forum (Proceedings Eurographics 2002)*, vol. 21, July 2002, pp. 595–604.

[21] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel, "Topological construction and visualization of higher order 3d vector fields," in *Computer Graphics Forum (Eurographics 2004)*, no. 3, October 2004, pp. 469–478.

[22] J. L. Helman and L. Hesselink, "Visualizing vector field topology in fluid flows," *IEEE Computer Graphics and Applications*, vol. 11, pp. 36–46, May 1991.
[23] G. Scheuermann, H. Krger, M. Menzel, and A. P. Rockwood, "Visualizing nonlinear vector field topology," *IEEE Transactions on Visualization and Computer Graphics*, vol. 4, no. 2, pp. 109–116, 1998.
[24] K. Polthier and E. Preuss, "Identifying vector fields singularities using a discrete hodge decomposition," in *Mathematical Visualization III*. Ed: H.C. Hege, K. Polthier, 2003, pp. 112–134.
[25] T. Wischgoll and G. Scheuermann, "Detection and visualization of planar closed streamline," *IEEE Transactions on Visualization and Computer Graphics*, vol. 7, no. 2, pp. 165–172, 2001.
[26] H. Theisel, T. Weinkauf, H. Hege, and H. Seidel, "Grid independent detection of closed stream lines in 2d vector fields," in *Proc. Vision, Modeling and Visualization 04*, 2004.
[27] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun, "Discrete multiscale vector field decomposition," in *ACM Transactions on Graphics (SIGGRAPH 03)*, vol. 22, no. 3, July 2003, pp. 445–452.
[28] T. Delmarcelle and L. Hesselink, "The topology of symmetric, second-order tensor fields," in *Proceedings IEEE Visualization 94*, 1994, pp. 140–147.
[29] H. Edelsbrunner, J. Harer, and A. Zomorodian, "Hierarchical morse-smale complexes for piecewise linear 2-manifolds," *Discrete Comput. Geom*, vol. 30, pp. 87–107, 2003.
[30] J. Palis, Jr. and W. de Melo, *Geometric theory of dynamical systems*. New York: Springer-Verlag, 1982, an introduction, Translated from the Portuguese by A. K. Manning.
[31] K. Mischaikow and M. Mrozek, "Conley index," in *Handbook of dynamical systems, Vol. 2*. Amsterdam: North-Holland, 2002, pp. 393–460.
[32] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to algorithms*. MIT Press, Cambridge, MA, 1990.
[33] K. Mischaikow and M. Mrozek, "Chaos in the Lorenz equations: a computer-assisted proof," *Bull. Amer. Math. Soc. (N.S.)*, vol. 32, no. 1, pp. 66–72, 1995.
[34] K. Mischaikow, "Topological techniques for efficient rigorous computation in dynamics," *Acta Numer.*, vol. 11, pp. 435–477, 2002.
[35] T. Kaczynski, K. Mischaikow, and M. Mrozek, *Computational homology*, ser. Applied Mathematical Sciences. New York: Springer-Verlag, 2004, vol. 157.
[36] M. S. Floater, "Mean value coordinates," *CAGD*, no. 20, pp. 19–27, 2003.
[37] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press, 1992.
[38] P. Alliez, D. Cohen-Steiner, O. Devillers, and M. Lévy, B.and Desbrun, "Anisotropic polygonal remeshing," in *ACM Transactions on Graphics (SIGGRAPH 03)*,, vol. 22, no. 3, July 2003, pp. 485–493.
[39] J. Wejchert and D. Haumann, "Animation aerodynamics," *Computer Graphics Proceedings, Annual Conference Series (SIGGRAPH 91)*, pp. 19–22, 1991.
[40] D. N. Kenwright, "Automatic detection of open and closed separation and attachment lines," in *Proceedings IEEE Visualization 98*, 1998, pp. 151–158.
[41] M. Eidenschink, "Exploring global dynamics : a numerical algorithm based on the conley index theory," Ph.D. dissertation, Georgia Institute of Technology, 1996.
[42] E. Zhang, J. Hays, and G. Turk, "Interactive tensor field design and visualization on surfaces," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 1, pp. 94–107, 2007.
[43] J. R. Cash and A. H. Karp, "A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides." *ACM Trans. Math. Softw.*, vol. 16, no. 3, pp. 201–222, 1990.
[44] 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*, 2nd ed., ser. Translations of Mathematical Monographs. Providence, RI: American Mathematical Society, 1986, vol. 66, translated from the Chinese by C. Y. Lo.
[45] X. Tricoche, C. Garth, G. L. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. D. Hansen, "Visualization of intricate flow structures for vortex breakdown analysis," in *Proceedings IEEE Visualization 04*, 2004, pp. 187–194.
[46] B. Jobard and W. Lefer, "Creating evenly-spaced streamlines of arbitrary density," in *EG Workshop on Visualization in Scientific Computing*, 1997, pp. 43–56.
[47] V. Verma, D. Kao, and A. Pang, "A flow-guided streamline seeding strategy," in *Proceedings IEEE Visualization 00*, 2000, pp. 163–170.
[48] T. Weinkauf, H. Theisel, K. Shi, H.-C. Hege, and H.-P. Seidel, "Extracting higher order critical points and topological simplification of 3d vector fields," in *Proceedings IEEE Visualization 05*, October 2005, pp. 559–566.
[49] R. Peikert and M. Roth, "The parallel vectors operator a vector field visualization primitive," in *Proceedings IEEE Visualization 99*, 1999, pp. 263–270.
[50] T. W. Hungerford, *Algebra*, ser. Graduate Texts in Mathematics. New York: Springer-Verlag, 1980, vol. 73, reprint of the 1974 original.

Guoning Chen received a BS degree in 1999 from Xi'an Jiaotong University and MS degree in 2002 from Guangxi University in China. He is pursuing a PhD in Computer Science at Oregon State University. He is currently working with Eugene Zhang on vector field analysis, visualization and design.

Konstantin Mischaikow received the B.A. in Mathematics from Reed College in 1979. He received the M.A. in Mathematics in 1983 and the PhD in Mathematics in 1985 from University of Wisconsin-Madison. He was a lecturer of University of Minnesota-Minneapolis from 1984 to 1985. During 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 recently awarded a PhD from the Vienna University of Technology, Austria at the Institute of Computer Graphics (2005). His research interests are in the areas of scientific visualization, computer graphics, and human-computer interaction. 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 department 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).

Paweł Pilarczyk received the MS in mathematics in 1997 and MS in computer science in 1999, both from the Jagiellonian University in Kraków, Poland. In 2001, he received the PhD degree in mathematics from the same university. In 1999-2006 he was an assistant professor at the Jagiellonian University, and in 2001-2003, as well as in 2005-2006 he was a visiting assistant professor at the Georgia Institute of Technology in Atlanta, USA. Since 2006, he is a JSPS Postdoc Fellow at the Department of Mathematics of the Kyoto University in Japan.

Eugene Zhang received the PhD degree in computer science in 2004 from 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, and computational topology. He is a member of the IEEE.