SCI INSTITUTE
TECHNICAL REPORT

2D Vector Field Simplification Based on Robustness

Primoz Skraba, Bei Wang, Guoning Chen, Paul Rosen

UUSCI-2013-004

Scientific Computing and Imaging Institute
University of Utah
Salt Lake City, UT 84112 USA

August 26, 2013

Abstract:

Vector field simplification aims to reduce the complexity of the flow by removing features in order of their relevance and importance, to reveal prominent behavior and obtain a compact representation for interpretation. Most existing simplification techniques based on the topological skeleton successively remove pairs of critical points connected by separatrices using distance or area-based relevance measures. These methods rely on the stable extraction of the topological skeleton, which can be difficult due to instability in numerical integration, especially when processing highly rotational flows. Further, the distance and area-based metrics are used to determine the cancellation ordering of features from a geometric point of view. Specifically, these metrics do not consider the flow magnitude, which is an important physical property of the flow. In this paper, we propose a novel simplification scheme derived from the recently introduced topological notion of robustness, which provides a complementary flow structure hierarchy to the traditional topological skeleton-based approach. Robustness enables the pruning of sets of critical points according to a quantitative measure of their stability, that is, the minimum amount of vector field perturbation required to remove them within a local neighborhood. This leads to a natural hierarchical simplification scheme with more physical consideration than purely topological-skeleton-based methods. Such a simplification does not depend on the topological skeleton of the vector field and therefore can handle more general situations (e.g. centers and pairs not connected by separatrices). We also provide a novel simplification algorithm based on degree theory with fewer restrictions and so can handle more general boundary conditions. We provide an implementation under the piecewise-linear setting and apply it to both synthetic and real-world datasets.

THE UNIVERSITY OF UTAH

2D Vector Field Simplification Based on Robustness

Primoz Skraba, Bei Wang, Guoning Chen, Paul Rosen

Abstract— Vector field simplification aims to reduce the complexity of the flow by removing features in order of their relevance and importance, to reveal prominent behavior and obtain a compact representation for interpretation. Most existing simplification techniques based on the topological skeleton successively remove pairs of critical points connected by separatrices using distance or area-based relevance measures. These methods rely on the stable extraction of the topological skeleton, which can be difficult due to instability in numerical integration, especially when processing highly rotational flows. Further, the distance and area-based metrics are used to determine the cancellation ordering of features from a geometric point of view. Specifically, these metrics do not consider the flow magnitude, which is an important physical property of the flow. In this paper, we propose a novel simplification scheme derived from the recently introduced topological notion of robustness, which provides a complementary flow structure hierarchy to the traditional topological skeleton-based approach. Robustness enables the pruning of sets of critical points according to a quantitative measure of their stability, that is, the minimum amount of vector field perturbation required to remove them within a local neighborhood. This leads to a natural hierarchical simplification scheme with more physical consideration than purely topological-skeleton-based methods. Such a simplification does not depend on the topological skeleton of the vector field and therefore can handle more general situations (e.g. centers and pairs not connected by separatrices). We also provide a novel simplification algorithm based on degree theory with fewer restrictions and so can handle more general boundary conditions. We provide an implementation under the piecewise-linear setting and apply it to both synthetic and real-world datasets.

Index Terms—vector field simplification, visualization, topological analysis, robustness, well groups.

1 Introduction

Vector field simplification aims to reduce the complexity of the flow by removing features in order of their relevance and importance, revealing prominent behavior and obtain a compact representation for interpretation. In addition, systematically simplifying the flow structure gives a consistent and multi-scale views of the flow dynamics.

A considerable amount of research has been focused on vector field simplification based on the notion of a topological skeleton [20, 22]. A topological skeleton consists of critical points connected by special streamlines called separatrices, which provides a condensed representation of the flow by dividing the domain into regions of uniform flow behavior. However, existing simplification techniques rely on the stable extraction of the topological skeleton, which can be difficult due to instability in numerical integration, especially when processing highly rotational flows, e.g. Figure 1. Furthermore, the distance and area-based relevance measures that are commonly used to determine the cancellation ordering of critical points typically rely on geometric proximity and do not consider the flow magnitude, which is an important physical property of the flow.

In this paper, we propose a new vector field simplification scheme derived from the recently introduced topological notion of *robustness*. Robustness, a notion related to *persistence* [14, 27], is used to represent the stability of critical points and thus assesses their significance with respect to perturbations of the vector fields. Intuitively, the robustness of a critical point is the minimum amount of perturbation that is required to cancel it within a local neighborhood, measured under an appropriate metric taking the flow magnitude into consideration. The contributions of our approach are:

- We propose a new simplification strategy based on robustness. Robustness enables the pruning of sets of critical points according to a quantitative measure of their stability. Such a strategy is (a) naturally hierarchical, (b) encodes flow magnitude, (c) gives a precise ordering of critical points cancellations, and (d) quantifies the amount of perturbation that is needed at each level.
- By comparing with the distance-based metric, we argue that the proposed strategy indeed provides a complementary view to the topological-skeleton-based simplification. The simplification does not depend on the topological skeleton nor require heuristic parameters. Therefore, (a) the method can handle more general situations (e.g. centers and pairs not connected by separatrices), and (b) the simplification can be computed efficiently based on sublevel set when separatrices are difficult to integrate.
- Our strategy is built on a novel simplification algorithm based on degree theory. The algorithm can handle more general boundary configuration without requirements on the Conley Index, compared to the existing methods when canceling pairs of critical points with opposite index. We provide an implementation under the piecewise-linear setting, and apply it to a number of synthetic and real-world datasets.

2 Related Work

Vector field simplification can be classified into topology-based and non-topology-based techniques [38]. Non-topology-based techniques

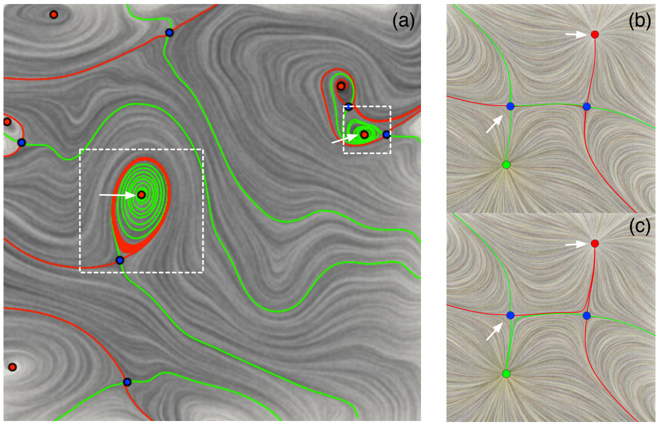

Fig. 1: Extraction of the topological skeleton. Sinks (and saddle-sink separatrices) are red, source (and saddle-source separatrices) are green, saddles are blue. (a) A highly rotational flow field where the pointed critical points are close to Hopf-bifurcations, such that numerical inaccuracies may accumulate during a long integration and different types of separatrices may intersect or switch. (b)-(c) Instability of separatrices under a small perturbation: The upper right sink is not connected with the saddle on the left in (b), but they are connected after a small perturbation to the vector field in (c).

- *Primoz Skraba is with Jozef Stefan Institute, Slovenija. E-mail: primoz.skraba@ijs.si.*
- *Bei Wang and Paul Rosen are with Scientific Computing and Imaging Institute, University of Utah. E-mails: {beiwang, prosen}@sci.utah.edu*
- *Guoning Chen is with University of Houston. E-mail: chengu@cs.uh.edu.*

typically focus on Laplacian smoothing of the potential of a vector field [24, 31, 37]. In particular, Tong et al. [31] proposed a multi-scale decomposition of a vector field using Hodge decomposition with each component smoothed independently before summing them.

Topology-based techniques modify the vector field topology explicitly by merging or cancelling nearby critical points based on the notion of topological skeleton [5, 11, 13, 20, 22, 33, 38]. One of the earliest investigations was done by De Leeuw and Van Liere [9, 10, 8]. They made use of a geometry-based relevance measure (e.g. with respect to distance or area proximity) to determine the pair of fixed points to be cancelled. A similar technique followed based on pair annihilations [33]. Tricoche et al. [32] focused on a piecewise analytic description for the simplified field, which was later extended to time-dependent 2D flows [34]. Theisel et al. [30] presented an algorithm for compressing vector fields while preserving their topology, and later combined it with simplification [29]. Zhang et al. [38] introduced a framework for fixed point pair cancellation based on Conley index theory for vector field editing. They extended this operation to critical point pairs that are *not* connected by a separatrix, such as a center and saddle pair. Chen et al. [5] extended this idea to include periodic orbits and presented a more complete pairwise cancellation framework.

Recently, a multi-scale hierarchy of the vector field topology has been defined and computed based on Morse decompositions [4, 6], where an explicit simplification technique based on this hierarchy is yet to be developed. Simplifications have also been proposed in a combinatorial setting [26, 28]. Edelsbrunner et al. [13, 14] performed 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. Finally, scale space techniques [21, 27] have also been proposed to assess the importance of a critical point for topology-based simplification.

In general, topological-skeleton-based methods focus on modification of a local region enclosing the critical points of interest while preserving the flow structure outside the region. There are different methods to obtain such a region. Zhang et al. [38] and Chen et al. [5] propose to compute an *isolating neighborhood* surrounding the critical points, where a modified vector field can be found by solving a constrained optimization problem. This method is particularly relevant in our context, as we will discuss it further in subsequent sections.

On the other hand, the notion of robustness is closely related to other measures derived from the concept of persistence [14], such as interval persistence [12], integral persistence [26], and scale-space persistence [27]. While persistence has been used successfully for scalar field visualization, robustness, first introduced in [15], is specifically designed for vector-valued data [3, 16]. Recent work [36] assigns robustness to critical points in both stationary and time-varying settings, and obtains a structural description of the vector field. Such a structural description implies the existence of a hierarchical simplification strategy based on robustness, which is the main focus of this paper.

3 Background

We provide relevant background in degree theory and robustness by reviewing previous work [3, 36] with minimal algebraic definitions and illustrating the related concepts through examples (Figure 2(a)-(b)) whenever possible. We also provide introductory descriptions of isolating neighborhoods and Laplacian smoothing used in vector field simplification [5, 38].

Degrees. The *(Poincaré) index* of a critical point x of a vector field, $\text{index}(x)$, is the number of field rotations while traveling along a closed curve centered at x counter-clockwise [35]. For a 2D vector field, we consider its first-order singularities, namely critical points such as sources, sinks, centers, and saddles. They have indices $+1$, $+1$, $+1$ and -1, respectively. A closely related concept is the *degree* of a continuous mapping (for its algebraic definition see [18] page 134 and [3]). For a critical point x in 2D, its degree $\deg(x)$ equals its index. Furthermore, for a (path-)connected component C in the domain that encloses several critical points, its degree $\deg(C)$ is the sum of the respective degrees of those critical points [3]. For our robustness-based simplification strategy, we rely on a corollary of the Poincaré-Hopf theorem (which is also employed by topological-skeleton-based simplification, e.g. [33]), which states that if a connected component C in 2D has degree zero, then it is possible to replace the vector field inside C with a vector field free of critical points.

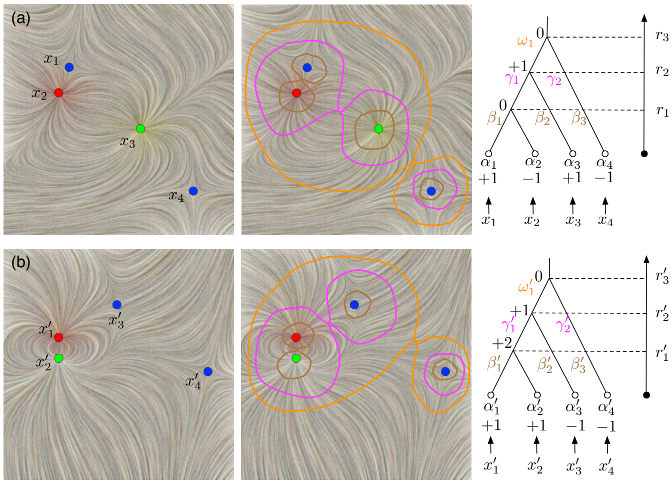

Fig. 2: Suppose the vector fields in examples (a) and (b) are continuous, where sinks are red, source are green and saddles are blue. From left to right: vector fields f, relations among components of $\mathbb{F}_r$, and augmented merge trees. (a) contains four critical points, a sink x_1, a source x_3, and two saddles x_2 and x_4. (b) contains four critical points, x'_1 (sink), x'_2 (source) , x'_3 and x'_4(saddles).

Merge tree. To analyze a continuous 2D vector field $f: \mathbb{R}^2 \to \mathbb{R}^2$, we define a corresponding scalar function $f_0: \mathbb{R}^2 \to \mathbb{R}$ which assigns for each point the magnitude (Euclidean norm) of the corresponding vector, $f_0(x) = ||f(x)||_2$. We use $\mathbb{F}_r = f_0^{-1}(-\infty, r]$ to denote the *sublevel set* of f_0 for some $r \geq 0$. We note that $\mathbb{F}_0$ is precisely the set of critical points of f – the points where the magnitude of the vectors goes to zero. A value $r > 0$ is a *regular value* of f_0 if for all sufficiently small $\varepsilon > 0$, $f_0^{-1}[r-\varepsilon, r+\varepsilon]$ and $f_0^{-1}(r)$ are diffeomorphic; otherwise it is a *critical value*. We assume that f_0 has a finite number of critical values and $\mathbb{F}_0$ is finite. We highlight the difference between critical points of f and the critical value of f_0.

Increasing r from 0, the space $\mathbb{F}_r$ evolves and we can construct a graph that tracks the (connected) components of $\mathbb{F}_r$ as they appear and merge. This is called a *merge tree* (or *join tree* [1]). The root represents the entire domain of f_0 while the leaves represents the creation of a component at a local minimum. An internal node represents the merging of two or more components. We further augment the merge tree with degree information at each node. That is, to each node, we record an integer that is the degree of the corresponding component in the sublevel set. Since the degree of a component is the sum of the degrees of the critical points lying in it, an initial computation of the degrees of critical points is sufficient to determine the degree of any component of any sublevel set [3].

Examples of such a tree is shown in Figure 2. We do not show any components which appear after $r = 0$ as they have zero degree and do not correspond to critical points of the vector field. We use α, β, γ, etc. to represent components of certain sublevel sets. In (a), the mapping $f_0: \mathbb{R}^2 \to \mathbb{R}$ has three critical values, denoted as $0 < r_1 < r_2 < r_3$. The merge tree on the right shows how the components of the sublevel sets $\mathbb{F}_r$ evolve. At $r = 0$ there are four components $\alpha_1, \alpha_2, \alpha_3$, and α_4 that correspond to the four critical points, each with non-zero degree. At $r = r_1$ there are three components in $\mathbb{F}_r$, two of which (β_2 and β_3) have non-zero degree. When $r = r_2$, components β_1 and β_2 merge into a single component γ_1 with non-zero degree, while β_3 grows into γ_2. Finally at $r = r_3$, the single component ω_1 has zero-degree. Similarly in (b), the degree information is augmented with the internal nodes of the merge tree as we track the evolution of components in $\mathbb{F}_r$ as r increases.

Static robustness and its properties. The *(static) robustness* of a critical point is the height of its lowest degree zero ancestor in the merge tree [2, 36]. The static robustness quantifies the stability of a critical point with respect to perturbations of the vector fields through the following lemmas explicitly stated in [36].

We first define the concept of *perturbation*. Let $f,h:\mathbb{R}^2\to\mathbb{R}^2$ be two continuous 2D vector fields. Define the distance between the two mappings as $\mathrm{d}(f,h)=\sup_{x\in\mathbb{R}^2}||f(x)-h(x)||_2$. A continuous mapping h is an *r-perturbation* of f, if $\mathrm{d}(f,h)\leq r$.

Lemma 3.1 (Critical Point Cancellation [36]) *Suppose a critical point x of f has robustness r. Let C be the connected component of $\mathbb{F}_{r+\delta}$ containing x, for an arbitrarily small $\delta>0$. Then, there exists an $(r+\delta)$-perturbation h of f, such that $h^{-1}(0)\cap C=\emptyset$ and $h=f$ except possibly within the interior of C.*

Lemma 3.2 (Degree and Critical Point Preservation [36]) *Suppose a critical point x of f has robustness r. Let C be the connected component of $\mathbb{F}_{r-\delta}$ containing x, for some $0<\delta<r$ and $r-\delta$ being a regular value. Then for any ε-perturbation h of f where $\varepsilon\leq r-\delta$, the sum of the degrees of the critical points in $h^{-1}(0)\cap C$ is* $\deg(C)$. *Furthermore, if C contains only one critical point x, we have* $\deg(h^{-1}(0)\cap C)=\deg(x)$. *In other words, there is no ε-perturbation ($\varepsilon\leq r-\delta$) that could cancel the critical point in C; that is, x is preserved.*

Now we revisit examples in Figure 2, by definition, the robustness of the critical points in (a), x_1, x_2, x_3, and x_4 are r_1, r_1, r_3, and r_3, respectively. Since the static robustness of x_1 is r_1, now for an arbitrarily small δ, consider a component $C\subseteq\mathbb{F}_{r_1+\delta}$ that is slightly larger than β_1 and contains x_1, then Lemma 3.1 implies the existence of an $(r_1+\delta)$ perturbation that cancels x_1 by locally modifying the component C. In example (b), the static robustness of all four critical points are r'_3. Let $C\subseteq\mathbb{F}_{r'}$ be a component that is slightly larger than γ'_1 containing x'_1 (where $r'_2<r'<r'_3$). By construction, $\deg(C)=+1$. Lemma 3.2 states that any r'-perturbation preserves the degree of C. In addition, since x'_4 has robustness r'_3, let $C'\subseteq\mathbb{F}_{r'}$ be a component slightly larger than γ'_2 containing only x'_4, for $r'_2<r'<r'_3$. Lemma 3.2 implies that no r'-perturbation will be able to remove x'_4.

Isolating neighborhood and Laplacian smoothing. We now review a distance-based simplification strategy that is based on isolating neighborhood, the Conley index and Laplacian smoothing. In such a simplification, we first locate pairs of critical points connected by separatrices and sort them in order of their Euclidean distance. If centers are present, we use the field rotated by 90 degrees to determine the pairing between saddles and centers [38]. For each pair of critical points, we compute its isolating neighborhood. Finally, we apply vector-valued Laplacian smoothing to remove high-frequency noise and reduce the number of critical points.

The concept of isolating neighborhood originates from Conley index theory. A region C is called *isolating* if all its boundary points can be uniquely classified as either an *entrance* (i.e. inflow) or an *exit* (i.e. outflow) point. In other words, C does not contain any inner tangencies along its boundary. If a region C in the domain contains multiple critical points and has trivial Conley index, then the associated theory tells us the flow inside C can be replaced with a new vector field free of critical points, e.g. using Laplacian smoothing [5, 38]. Here we ignore the technical definition of Conley index [7], and only mention that a typical situation for C to have trivial Conley index is when its boundary ∂C contains a single inflow and a single outflow component.

To compute an isolating neighborhood that satisfies the trivial Conley index requirement, existing methods [5, 38] proceed as follows. Consider a pair of critical points connected by a separatrix, one of them is referred to as a *repeller* while the other an *attractor* based on the separatrix direction that connects them. In the piece-wise linear (PL) setting, a region, C^+, is grown from the triangle containing the repeller in the forward direction of the flow. At the same time, another region, C^-, is grown from the triangle containing the attractor in the backward flow direction. The intersection of these two regions, $C=C^+\cap C^-$, gives rises to the desired isolating neighborhood.

Given a vector field f and an isolating neighborhood C, a modified vector field $\overline{f}$ inside C can be found by solving a constrained optimization problem, referred to as *Laplacian smoothing*. Specifically, a vector-valued discrete Laplacian equation is solved over C in the domain (e.g., a triangular mesh) where the vector values at the boundary vertices of C are fixed. We employ the equation

$$\overline{f}(v_i)=\sum_j\omega_{ij}\overline{f}(v_j),$$

where v_i is an interior vertex and v_j's are the adjacent vertices that are either in the interior or on the boundary of C. The weights ω_{ij} are usually determined using Floater's mean-value coordinates [17]. This is a sparse linear system, which can be solved using a conjugate gradient method [25]. Although this framework typically performs well in practice, there is no guarantee that a critical point free field can always be found due to the method that is employed to solve for the linear system and, more importantly, the spatial discretization.

4 Robustness-Based Simplification Algorithms

In robustness-based simplification, we first locate sets of critical points that share the same lowest zero-degree ancestors in the merge tree and sort them based on their robustness values. For each such set with shared robustness value r, we compute the corresponding component of the sublevel set $C\subseteq\mathbb{F}_r$ that contains it. Since by construction $\deg(C)=0$, we can apply our simplification strategy to remove the critical points in C.

While the distance-based strategy applies simplification to an isolating neighborhood with trivial Conley index, the robustness-based strategy applies simplification to components in the sublevel set of the flow magnitude function with zero degree. For the remaining of this section, we introduce relevant concepts and algorithmic procedures with intuitive illustrations and synthetic examples whenever possible.

4.1 Preliminary

In this section, we introduce the relevant concepts and constructions. First we consider them in a smooth setting, and then translate the corresponding language into the PL setting.

Given a 2D vector field restricted to the degree-zero component C, $f:C\to\mathbb{R}$, we define the *image space* of C, denoted as $\mathrm{im}(C)$. For each point $p\in C$, we have a vector $v_p=f(p)\in\mathbb{R}^2$. $\mathrm{im}(C)\subset\mathbb{R}^2$ is constructed by mapping a point p to its vector coordinates v_p. The origin in $\mathrm{im}(C)$ corresponds to the critical points (0-magnitude vectors) in C. Since $C\subseteq\mathbb{F}_r$, it follows that $\forall p\in C$, $||v_p||_2\leq r$, therefore $\mathrm{im}(C)$ is contained within a 2D disk of radius r in $\mathbb{R}^2$. We denote the boundary of such a disk as S.

Now suppose the boundary of C, denoted as ∂C, is a simple closed curve[1]. Note that the above mapping maps ∂C to S, obtaining its image as $\mathrm{im}(\partial C)$. We refer to the boundary of $\mathrm{im}(C)$ to be *uncovered*, if $\mathrm{im}(\partial C)\subset S$, otherwise, it is *covered*. Figure 3(a)-(b) illustrate these concepts. Note that both examples have zero degree. In 3(a), the region C encloses a saddle-sink pair connected by separatrix. By traversing counter-clockwise along ∂C and observing how its image $\mathrm{im}(\partial C)$ wraps around S, we see that the boundary of $\mathrm{im}(C)$ is uncovered. In 3(b), the region C encloses a saddle-sink pair not connected by separatrix, and the boundary of $\mathrm{im}(C)$ is covered.

In the PL setting, the vector field f is restricted to a triangulation K of C, $f:K\to\mathbb{R}^2$, where the support of K, $|K|=C$. We construct the image of C by mapping each vertex $p\in K$ to its vector coordinates $v_p=f(p)$. Through linear interpolation, this also maps edges and triangles in K to edges and triangles in $\mathrm{im}(C)$. This process is illustrated in Figure 4. The concept of *covered* and *uncovered* boundaries of $\mathrm{im}(C)$ can be defined similarly up to a small additive constant.

4.2 Algorithm Overview

Our simplification strategy consists of three operations which are applied to C (while keeping its boundary fixed):

- **Smoothing**(C): Perform Laplacian smoothing on C (as described in Section 3).

[1] This is not strictly needed, but it simplifies the algorithm and exposition.

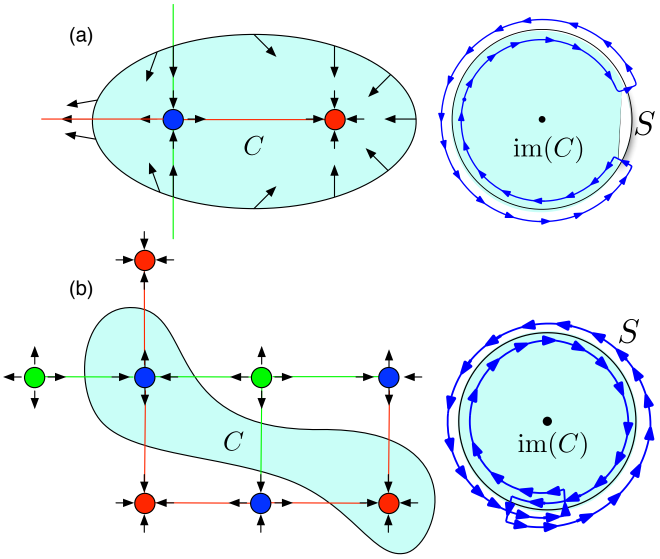

Fig. 3: Illustrative examples for uncovered (a) and covered (b) boundaries of im (C).

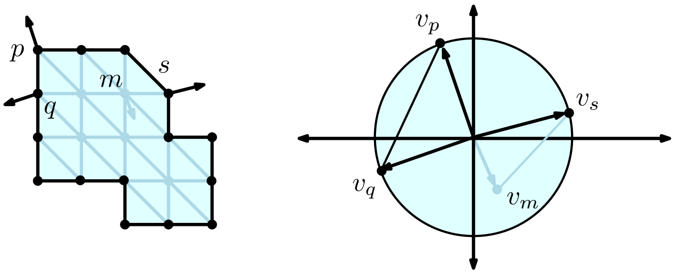

Fig. 4: An example component and its image space under the PL setting where a few vector mappings are highlighted.

- **Cut**(C): Deform the vector field in its image space im (C) to remove critical points in C.
- **Unwrap**(C): Modify the vector field in its image space im (C) such that part of its boundary is uncovered.

There are three possible cases which are classified by the Conley index of C, denoted as CH$_*(C)$. The operations which are applied to simplify each case are:

- **Case (a)**: If CH$_*(C)$ is trivial, return $C_1 =$ **Smoothing**(C).
- **Case (b)**: If CH$_*(C)$ is non-trivial and the boundary of im (C) is uncovered, then $C_1 =$ **Cut**(C), and return $C_2 =$ **Smoothing**(C_1).
- **Case (c)**: If CH$_*(C)$ is non-trivial and the boundary of im (C) is covered, then (i) $C_1 =$ **Unwrap**(C), (ii) $C_2 =$ **Cut**(C_1) and return $C_3 =$ **Smoothing**(C_2).

By construction, $deg(C) = 0$ in all 3 cases. Indeed, $deg(C) \neq 0$ is a sufficient condition such that there exists no simplification. As we go from the most restrictive case (a) to the least restrictive case (c), additional processing is required, as detailed in the next section.

4.3 Algorithm Details

We now describe the **Cut** and **Unwrap** operations in detail. Note that **Smoothing** is an operation we employ to achieve visually appealing results, we focus on **Cut** and **Unwrap** operations and discuss the maximum amount of perturbation needed due to these operations.

Cut operation. Suppose the boundary of im (C) is uncovered. The idea behind the **Cut** operation is to deform the vectors in im (C) such that there is a small neighborhood surrounding the origin that is not covered by im (C). This corresponds to the situation where there is no critical point in C after the deformation and is illustrated in Figure 5.

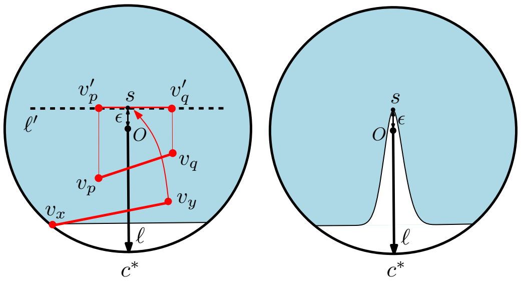

Fig. 5: **Cut** operation. Left: The projection of edges which intersect ℓ during the **Cut** operation. Right: After **Cut**, the light blue region represents im (C) which no longer contains (covers) the origin and so is critical point free.

As shown in Figure 5 left, first we pick a vector, whose initial point is located at the origin O and its terminal point is at an arbitrary point c^* from the uncovered part of the circle S (we refer to such a point as the *cut point*). Let ℓ be a line that coincides with such a vector passing through c^*. Define another line ℓ' which is orthogonal to ℓ and is ε away from the origin. The point $s \in \ell'$ is at a distance ε from the origin. Next, we find all the mesh edges $v_p v_q$ (corresponding to the edge pq in K) in the interior of im (C) that intersect with the line ℓ, and project their end points along the direction of ℓ onto ℓ', forming the projected edge $v'_p v'_q$. In the original domain, the vectors at $p, q \in K$ are deformed from v_p and v_q to the vectors v'_p and v'_q respectively. Third, we locate all the mesh edges $v_x v_y$ where $x \in \partial C$ (and so v_x is on the boundary of im (C)) and v_y is in the interior. We move the point v_y to s so that the edge $v_x v_y$ no longer crosses ℓ and the boundary vector remains unchanged. Since the boundary of im (C) is uncovered, there is no edge that intersects ℓ whose end points are both located on the boundary of im (C) (i.e. whose corresponding points are both in ∂C).

The above operation therefore creates an empty wedge around the origin (as shown in Figure 5 right), which ensures that there is no critical points in C after the modification. By construction, the amount of perturbation is less than $r + \varepsilon$.

The procedure to find a cut point c^* is illustrated in Figure 6(a)-(b). In (a), by traversing counter-clockwise along ∂C and observing how its image im (∂C) (blue curve) wraps around S, we define the angle θ of a point along S to be its *phase*. In (b), we showcase (in blue) the corresponding phase plot (a.k.a. angle-valued function), that is, a function $h : \partial C \rightarrow \theta$ where $\theta \in [-\pi, \pi]$. Traversing ∂C again, we can use *phase-unwrapping* to give us a continuous function $\varphi : \partial C \rightarrow \phi$ for $\phi \in \mathbb{R}$ (shown in red) using the following equation

$$\varphi(i) = \left\lfloor \theta(i) - \phi(i-1) + \frac{1}{2} \right\rfloor + \theta(i).$$

Since the boundary of im (C) is uncovered, it follows that $\max_{\partial C}(\varphi) - \min_{\partial C}(\varphi) < 2\pi$. We set the cutting angle ϕ^* as the mid-point of the uncovered part and recover the corresponding cut point on S by

$$\begin{aligned} \phi^* &= \frac{\max_{\partial C}(\varphi) + \min_{\partial C}(\varphi)}{2} + \pi, \\ c^* &= (r \cos \phi^*, r \sin \phi^*), \end{aligned}$$

where r is the robustness parameter of the sublevel set (and the diameter of the disk in the image space). By finding this cut point c^* using the phase parameter θ, we do not have to worry about the PL effects in determining where the boundary is uncovered.

Unwrap operation. Suppose the boundary of im (C) is covered. In this case, we must first **Unwrap** the boundary so that part of it becomes uncovered, then we perform the **Cut** procedure and finally restore the boundary.

The **Unwrap** operation is divided into several steps illustrated in Figure 6(c)-(e). We must first determine the optimal *unwrap point*. This step is similar to determining the *cut point*. As before, we traverse ∂C and compute a phase plot (a.k.a. an angle-valued function) h :

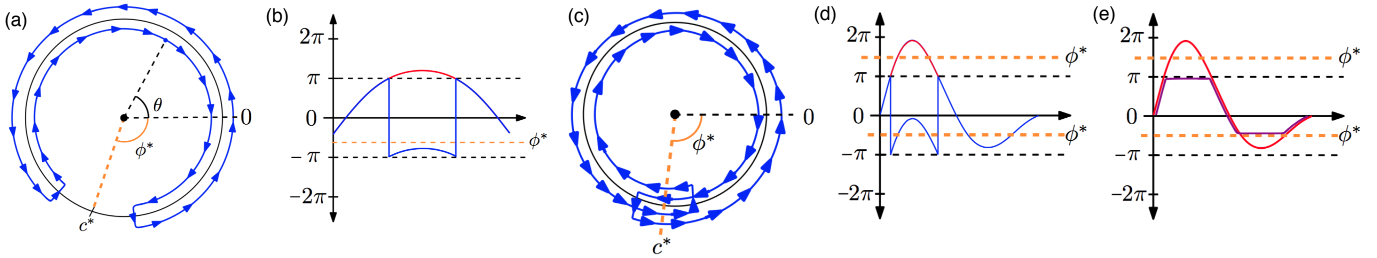

Fig. 6: (a)-(b) Locating a cut point for the **Cut** operation: (a) Track the angle (a.k.a. phase) of a point in $\text{im}(\partial C)$ along S as we move along ∂C counter-clockwise. (b) The corresponding phase plot (a.k.a. angle-valued function) is shown in blue. The result of phase-unwrapping is shown in red. (c)-(e) Locating an unwrap point in **Unwrap** operation: (c) Track the angle of a point in $\text{im}(\partial C)$ along S as we move along ∂C counter-clockwise. (d) The corresponding angle-valued function (shown in blue), the result of phase-unwrapping (shown in red), and the optimal unwrap point c^* corresponding to phase ϕ^*. (e) The modified boundary of $\text{im}(C)$ (shown in purple) which becomes uncovered.

$\partial C \to \theta$, and apply phase-unwrapping to obtain a continuous function $\varphi : \partial C \to \phi$ (Figure 6(c)-(d)). The unwrap point c^*, is defined by

$$\phi^* = \frac{\max_{\partial C}(\phi) + \min_{\partial C}(\phi) + 2n\pi}{2},$$

where n is the smallest integer such that $|\min(\theta) + 2n\pi - \max(\theta)| < \pi$, and $c^* = (r\cos\phi^*, r\sin\phi^*)$.

To **Unwrap** the boundary, let $X \in \partial C$ be the set of points on the boundary such that $\phi(X) > \phi^* - \delta$, and $Y \in \partial C$ be the set of points on the boundary such that $\phi(Y) < \phi^* + \delta - 2n\pi$. To **Unwrap** we set

$$\phi(X) = \phi^* - \delta, \qquad \phi(Y) = \phi^* - 2n\pi + \delta,$$

as illustrated in Figure 6(e) and set the corresponding vectors to

$$\begin{array}{rcll} v_x & = & (r\cos(\phi(x)), r\sin(\phi(x))) & x \in X \\ v_y & = & (r\cos(\phi(y)), r\sin(\phi(y))) & x \in Y \end{array}$$

where r is the magnitude of the vectors on the boundary (e.g. the sublevel set parameter). The final step is to restore the boundary to its original values, which again covers the boundary but keeps the origin

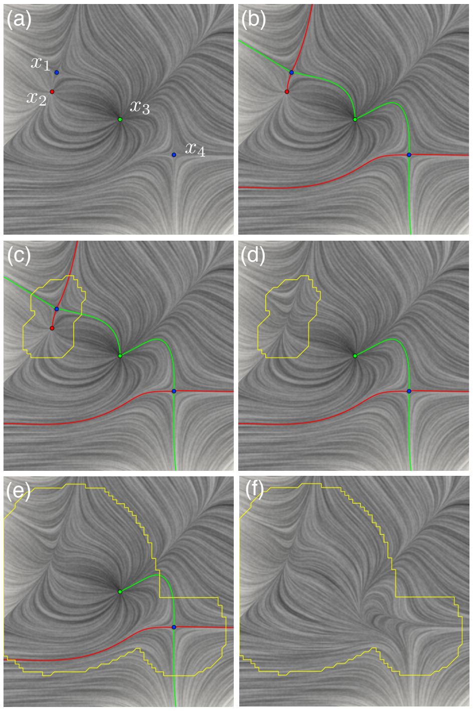

Fig. 7: SyntheticA. (a) The original vector field, sinks are red, sources are green and saddles are blue. (b) The topological skeleton, saddle-sink separatrices are red, saddle-source separatrices are green. (c)-(d) 1st level simplification: before (left) and after (right) **Smoothing**. (e)-(f) 2nd level simplification: before (left) and after (right) **Smoothing**.

uncovered. While not as obvious as in case (b), the deformation is again bounded by $r + \varepsilon$. We omit the proof here, but the key observation is that we only move internal nodes by less than $r + \varepsilon$. While the unwrapping may introduce larger deformations, it allows us to determine how to deform the internal vector field, as the output has its boundary restored (e.g. re-wrapped). This procedure is showcased in the next section with the SyntheticC example.

4.4 Synthetic Examples

We illustrate our robustness-based simplification strategy on three PL synthetic examples, highlighting the three different cases.

SyntheticA (Figure 7) corresponds to the example in Figure 2(a). It involves paris of critical point connected by separatrices. At r_1, we have a component which contains critical points x_1 and x_2 and at r_2 we have a component which contains all four critical points x_1 to x_4. The simplification hierarchy involves two steps ranked by robustness values: we first simplify x_1 and x_2 then simplify x_3 and x_4. Since both components (marked by yellow boundary) have trivial Conley index, their simplifications correspond to case (a), where only **Smoothing** operations are needed.

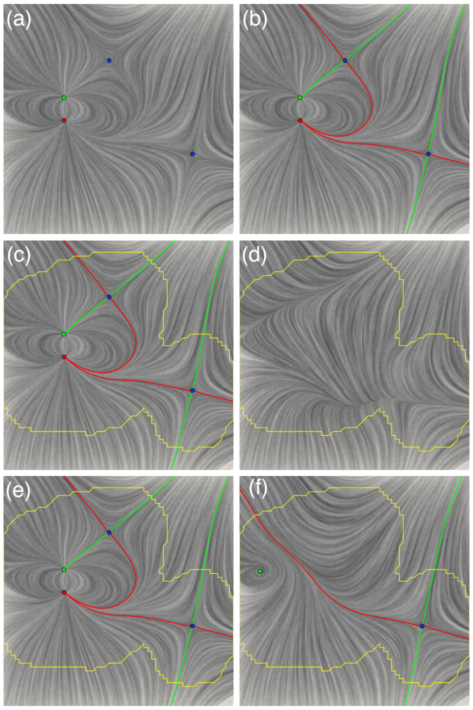

Fig. 8: SyntheticB. (a) The original vector field. (b) The topological skeleton. (c)-(d) Single level simplification before (c) and after (d) by combining **Cut** and **Smoothing**. (e)-(f) Only applying **Smoothing** does not make the region a critical point free field.

SyntheticB (Figure 8) corresponds to example (b) in Figure 2. It involves a group of four critical points which are interconnected by sep-

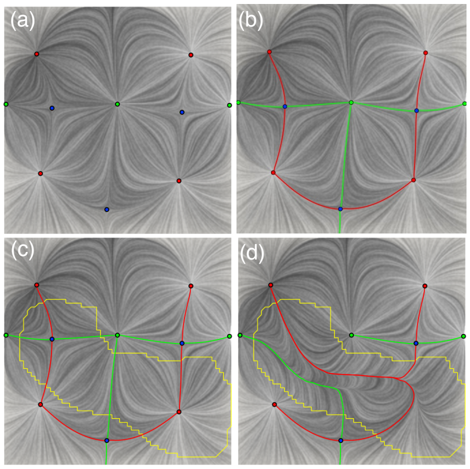

Fig. 9: SyntheticC. (a) The original vector field. (b) The topological skeleton. (c)-(d) Single level simplification before (c) and after (d) by combining **Unwrap**, **Cut** and **Smoothing**.

aratrices, which could be simplified in a single level using robustness-based strategy. Since the component of interest has non-trivial Conley index, directly applying Laplacian smoothing fails (as shown in Figure 8 (e)-(f)). The component's boundary is uncovered, so we apply case (b) of our simplification by combining **Cut** with **Smoothing**.

SyntheticC (Figure 9) corresponds to case (c) of our algorithm. This is an untypical case involving a pair of critical points not directly connected by a separatrix. In this case, the component of interest C has non-trivial Conley index, and the boundary of its image is covered. The robustness-based strategy cancels the critical point pair without any issue by combining **Unwrap**, **Cut** and **Smoothing** operations. We further focus on this example by illustrating the image space of C, $\mathrm{im}(C)$, during various steps of simplification in Figure 10. In Figure 10(a), the entire boundary and disk are covered. However, from the left phase plot in Figure 11, we can see that the degree is 0. Once the optimal unwrapping point is computed, we perform the **Unwrap** operation, giving the right phase plot in Figure 11 and the image space in Figure 10(b), leaving the boundary S uncovered. The effect of the **Cut** operation in image space is shown in Figure 10(c), creating a void surrounding the origin. Lastly, in Figure 10(d), the boundary is restored to its original value while leaving the origin uncovered.

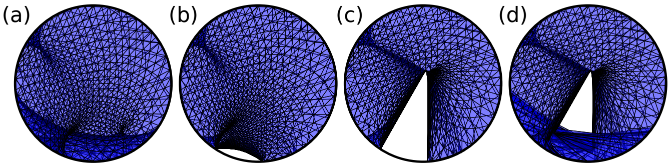

Fig. 10: SyntheticC. The image space is shown through the different steps: (a) original, (b) after **Unwrap**, (c) after **Cut** and (d) final output.

5 Results

We demonstrate our robustness-based simplification strategy on a number of real-world datasets. Whenever possible, we compare our simplification hierarchy and results with distance-based simplification.

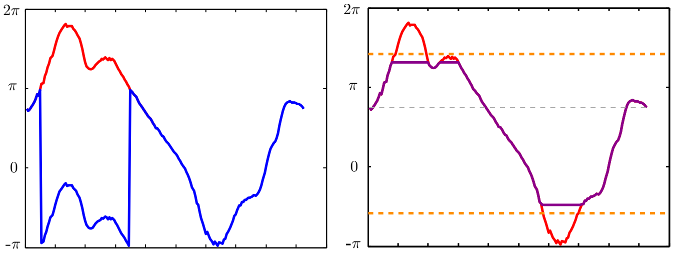

Fig. 11: SyntheticC. Left: The phase plot, original version (blue) and the phase-unwrapped version (red). Right: The phase plot with optimal unwrap point (orange) and the modified phase plot with boundary uncovered (purple).

The first real-world dataset we explore is the simulation of global oceanic eddies [23] for 350 days of the year 2002. We consider the top layer of the 3D simulation which is represented as a 2D time-varying vector field of resolution 3600 × 2400. We extract tiles from this simulation data, representing the flow in the central Atlantic Ocean (60 × 60) and construct standard triangulation on the point samples. We select multiple time slices from this data: OceanA contains slices #21217 and #21311; OceanB and OceanC corresponds to slices #20904 and #20821, respectively; OceanD includes a time-varying sequence of slices from #20710 to #20715.

Our second real-world dataset is taken from the simulation of homogeneous charge compression ignition (HCCI) engine combustion [19]. The domain has periodic boundary and is represented as a 640 × 640 regular grid. The 2D time-varying vector field consists of 299 time-steps with a time interval of 10^{-5} seconds. We selected slice #173 from this data, referred to as the Combustion dataset.

5.1 Topologically Equivalent Scenarios

In many scenarios, our approach produces topologically equivalent results to the distance-based approach. The resulting modifications to the vector field may differ; however, the obtained simplified vector fields are topologically identical using both approaches. OceanB dataset (Figure 12(a)) provides an example where both approaches agree. The two critical point pairs of interest are highlighted by the black dashed boxes in the top row left. Here critical points are colored by their robustness values, with red being low and white being high robustness. The upper right pair has higher robustness than the lower middle pair, and is further apart in distance. The simplification results generated by distance-based and the robustness-based approach are shown in the second and third row respectively. The approximated isolating neighborhoods are highlighted by the white boxes (middle row) while the sublevel sets the yellow enclosure (bottom row). From the comparison, we observe that: First, both the distance and robustness-based metric generate the same pairs of critical points. Second, the simplification orderings determined by these two metrics agree. However, a subtle difference in the resulting vector fields from the two approaches is visible due to the different local regions determined by the two metrics and different algorithms for modifying the local vector fields. Note that, the local regions determined by the computation of isolating neighborhood are typically larger than those derived from sublevel set of robustness value. Therefore, even though the two approaches generate topologically equivalent results, the robustness-based method requires smaller change to the original vector field.

OceanA dataset (Figure 13) shows a more complex scenario where the local region encloses more than two critical points, i.e. a cluster of critical points. The two vector fields in this example are from slices #21217 and #21311. Each of these two clusters (highlighted by the black dashed boxes in Figure 13 top row) consists of four critical points that are close in distance and have small identical robustness values. The robustness metric groups them as one cluster automatically and computes a region based on their sublevel set. The bottom row of Figure 13 provides the simplification results using the algorithm introduced in Section 4. Although the distance-based method cannot group these four critical points in one simplification, for comparison purpose we compute an isolating neighborhood that enclose them. Laplace smoothing is then performed in this local region (Figure 13 middle row). From the comparison, we see that both methods return similar results. Nevertheless, the robustness-based method can apparently handle regions with more complex boundary configurations.

5.2 Inconsistent Hierarchical Scenarios

We also identified a number of scenarios where the distance-based and robustness-based approaches do not agree. OceanC dataset (Figure 12 (b)) provides such an example. In this dataset, two pairs of critical points are studied (highlighted in the top row of Figure 12 (b)). Even though these four critical points are paired consistently using both metrics, their actual simplification orderings are different. The distance-

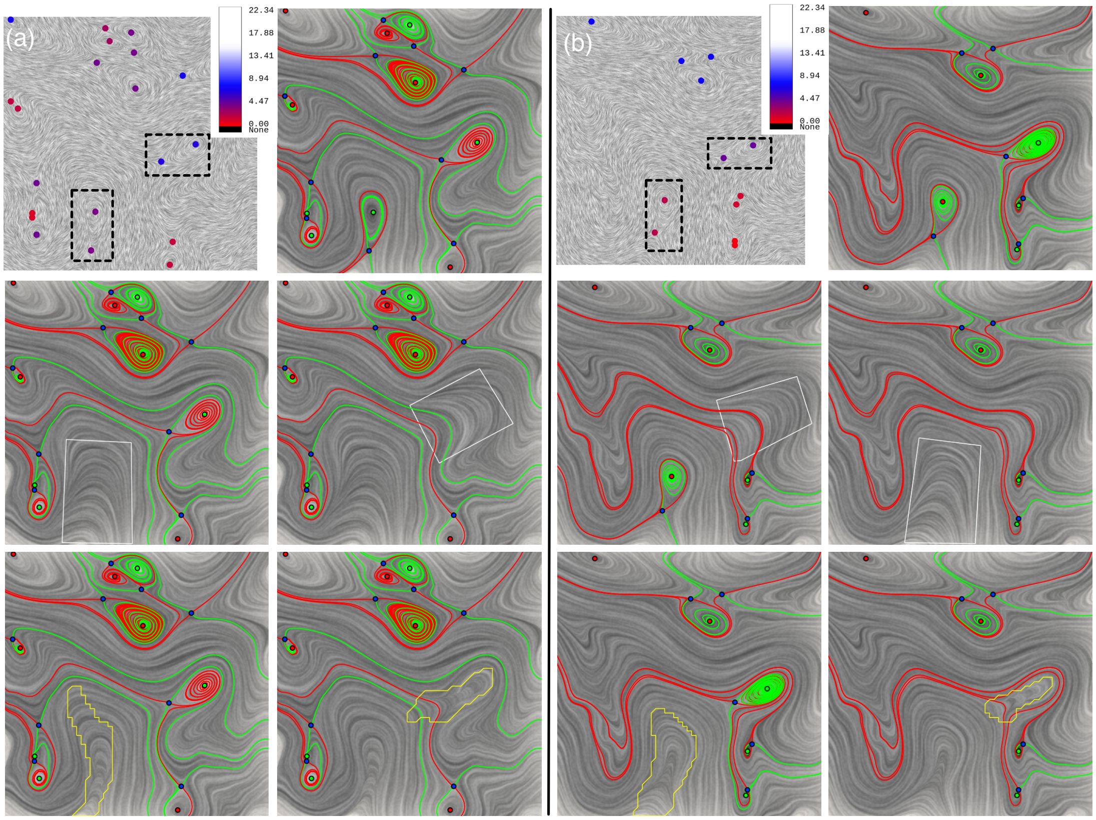

Fig. 12: (a) The OceanB dataset. (b) The OceanC dataset. For each subfigure: Top Row: Left – shows robustness values with the region of interest highlighted, robustness values are colored from red to white, where red means low and white means high robustness; Right – shows the vector field marked by critical point types along with separatrices. Middle Row: the two-step hierarchical simplification based on distance. Bottom Row: the two-step hierarchical simplification based on robustness.

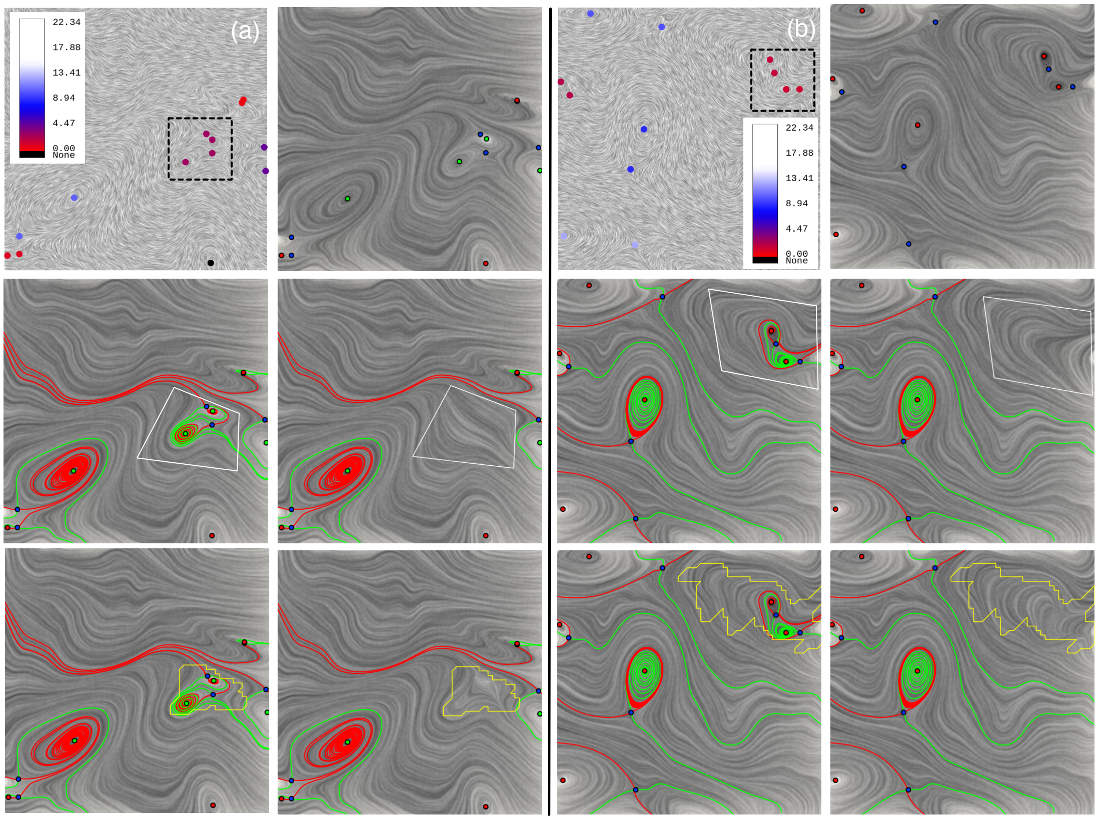

Fig. 13: The OceanA dataset: (a) #20311; (b) #21217. For each subfigure, Top Row: Left – shows robustness values with region of interest highlighted; Right – shows the vector field marked by critical point types. Middle Row: before (left) and after (right) distance-based simplification. Bottom Row: before (left) and after (right) robustness-based simplification.

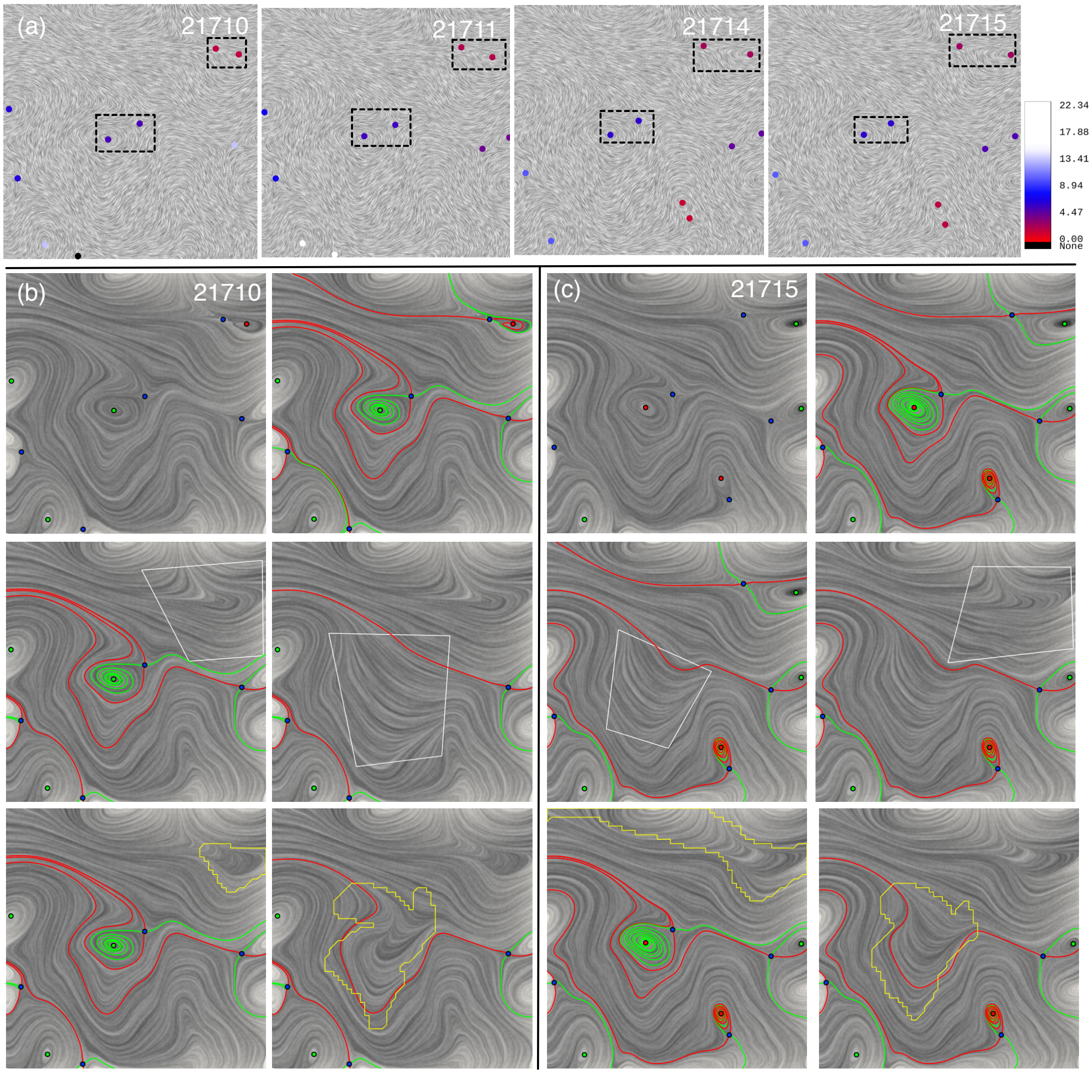

Fig. 14: The OceanD dataset. (a) A sampled time serie with pairs of critical points highlighted. (b) #21710. (c) #21715. For each subfigure (b)-(c), Top Row: The original vector field (left) and with (right) the separatrices. Middle Row: The simplification ordering for the distance-based strategy. Bottom Row: The simplification ordering for the robustness-based strategy. Orderings for distance and robustness-based methods are consistent in (b) and different in (c).

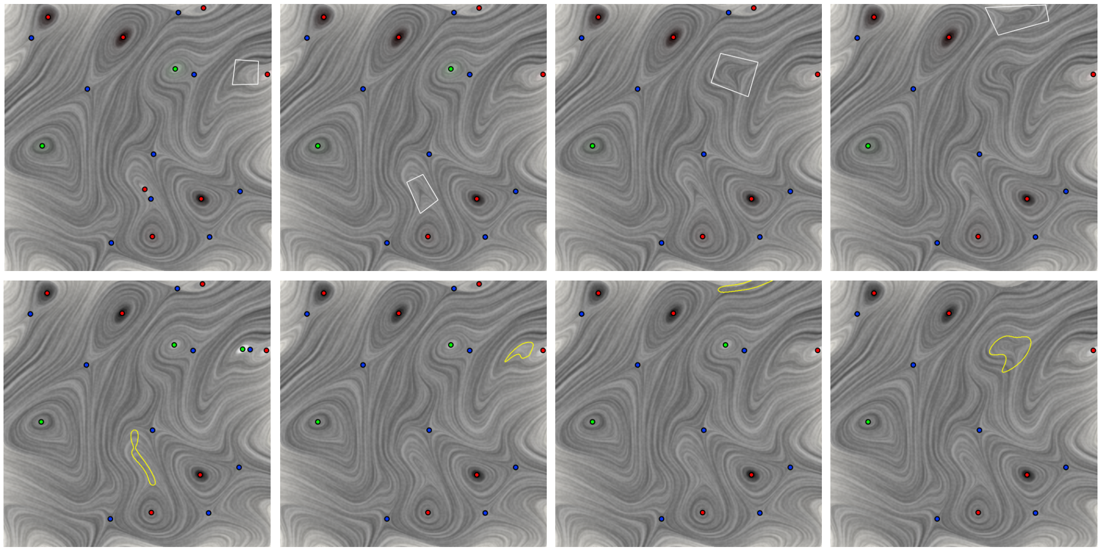

Fig. 15: The Combustion dataset. Top: The bottom-up hierarchical simplifications from the distance-based strategy. Bottom: The bottom-up hierarchical simplifications from the robustness-based strategy.

based method cancels the pair in the middle-right of the domain first, while the robustness-based method cancels the lower-middle pair first. Figure 14 provides another example that shows the discrepancy of the two approaches in determining the simplification ordering of critical point pairs in the time-varying setting. In this example, we look at a number of consecutive time steps from the OceanD dataset. The top row of Figure 14 highlights the pairs of the critical points that we are interested in. The pairing of these four critical points again agree with each other using both topological-skeleton and robustness metric. We perform a per-slice simplification using the two approaches. The results are shown in the third (distance-based) and fourth (robustness-based) rows, respectively. From the results, we see that the cancellation orderings are changing over time using the distance-based metric. This is because the distance between the two critical points near the upper-right corner is getting larger, while the other pair is getting closer over time, resulting in the change of the simplification ordering. On the other hand, the robustness values for these two pairs are stable during the time evolution. Therefore, the robustness-based simplification returns consistent outcome. This examples shows that the robustness-metric can capture physical properties of the flow dynamics (i.e. flow magnitude) and provide a more consistent simplification over time.

5.3 Challenging Scenarios

There are a number of cases where the topological-skeleton based metric combined with the Laplace smoothing technique is incapable of simplifying the given vector field. SyntheticB dataset shown in Figure 8 is such an example. In this case, it is impossible to find an isolating neighborhood with trivial Conley index that encloses all the critical points due to the boundary condition. Therefore, even though the obtained local region is guaranteed to be zero-degree, the Laplace smoothing fails to solve for a critical point free field. On the other hand, the simplification algorithm introduced in Section 4 successfully simplifies the field. A similar situation occurs in the example shown in Figure 16 (OceanA slice #21217). In this example, we try to apply Laplace smoothing in the local region computed based on the sublevel set of the robustness value of a cluster of critical points (left). The boundary configuration of this region is rather complex and does not satisfy trivial Conley index. The Laplace smoothing based on this boundary configuration fails, while the proposed simplification method succeeds. These two examples showcase the utility of the proposed simplification algorithm in solving a critical point free field within any given regions with zero degree. This relieves the requirement of the trivial Conley index whose corresponding isolating neighborhood is sometimes difficult to obtain.

Figure 9 illustrates a non-typical case that involve the cancellation of a pair of critical points not directly connected by separatrix. It is impossible for the topological-skeleton based method to compute an isolating neighborhood that encloses two critical points not connected by separatrix [5]. Nonetheless, the robustness metric is able to derive a local region that encloses only these two critical points with total degree equal to zero under certain configuration of the flow magnitude. Hence, these two critical points can be cancelled using the proposed algorithm. While this may not occur often in the real-world data, it illustrates the flexibility and generality of the proposed method. In practice, a simpler but similar situation may occur.

Figure 15 shows one slice from the combustion data. The corresponding vector field is an incompressible flow. Therefore, the conventional topological-skeleton based method cannot be directly applied as the separatrices either do not exist or are difficult to integrate correctly. Additional step will be required, for example, rotating the field by 90 degree and computing the topological-skeleton for the obtained dual vector field in order to pair the critical points. In addition, this dataset is much larger than the synthetic and ocean data sets, computing topological-skeleton for the dual vector field and deriving the subsequent isolating neighborhood for a given pair is computationally expensive. In contrast, the robustness-based method does not require the computation of the topology, and its sublevel set computation is fast and can be augmented using parallel computation. Therefore, it could be more suitable for the processing of large-scale datasets. Figure 15 compares the simplification results of the different ordering determined by the distance-based metric (first row) and the robustness-based metric (second row), where their corresponding bottom-up hierarchies do not agree.

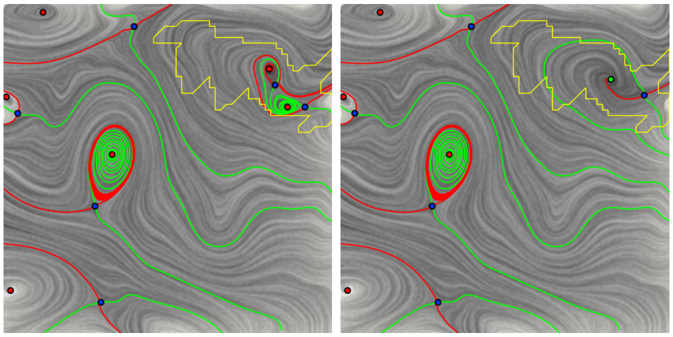

Fig. 16: OceanA slice #21217. A region (enclosed by yellow boundary) with non-trivial Conley index and uncovered boundary, where direct smoothing operation does not remove its critical points.

6 Discussions

We have presented a new and complementary simplification framework based on the robustness metric. It does not depend on the topological skeleton. It does, however, incorporate topological information through *robustness* – a generalization of topological persistence. Rather than considering the distance proximity between critical points, the intuition behind robustness-based simplification is to consider the maximum amount of perturbation we must apply to any one vector to remove critical points.

The simplification algorithm comes with theoretical guarantees on the bounds on the amount of perturbation we introduce, whenever cutting and/or unwrapping is used. The main motivation for introducing Laplacian smoothing is to produce more visually appealing results. However, to the best of our knowledge, no bounds exist on the amount of perturbation based on Laplacian smoothing, other than the obvious maximum distance between points on the boundary. In the context of robustness, this is twice the amount of perturbation our simplification method alone uses. In practice, the addition of Laplacian smoothing does increase the amount of perturbation but not significantly.

Our procedure provides us a natural simplification hierarchy and works in a wide variety of settings.

Scalability: Our method is very efficient and should scale to very large datasets. The computation of the robustness hierarchy is based on a merge tree computation that runs in near linear time. This encodes all the information we require for computing the simplification – there is no need for a separate integration of separatrices. Determination of the cut or unwrap point is done by only considering the boundary of the region to be simplified. In case the region of interest has non-zero degree, the algorithm first detects such a condition (specifically, during the phase-unwrapping) and then simplifies the region as much as possible.

Generality: The simplification procedure only requires that the degree of the boundary be zero and so applies to a wide range of cases. It can deal with highly rotational data (e.g. centers) as well as cases where critical points are not connected by separatrices. Indeed, we have shown that some unusual, non-intuitive simplifications may not only exist but can be efficiently computed. We believe these methods allow us to handle much more general cases and have much less restrictive conditions on the vector field data to be analyzed.

Extensions: While our simplification hierarchy is built on robustness and sublevel sets of magnitude function, we note that the actual simplification only requires degree-zero component. Therefore, we could consider other metrics for constructing hierarchies such as incorporating both the magnitude of the vectors and the area, to capture a quantity closer to the energy of a perturbation. The algorithm would require some modification, but the use of the phase plot to determine the cut or unwrap point is agnostic to the metric being used. Of course it remains an open question what other methods exist to generate degree-zero regions for simplification.

REFERENCES

[1] H. Carr, J. Snoeyink, and U. Axen. Computing contour trees in all dimensions. *Proceedings of ACM-SIAM Symposium of Discrete Algorithms*, pages 918–926, 2000.

[2] F. Chazal, A. Patel, and P. Skraba. Computing the robustness of roots. Manuscript, http://ailab.ijs.si/primoz_skraba/papers/fp.pdf, 2011.

[3] F. Chazal, A. Patel, and P. Skraba. Computing well diagrams for vector fields on R^n. *Applied Math. Letters*, 25(11):1725–1728, 2012.

[4] G. Chen, Q. Deng, A. Szymczak, R. Laramee, and E. Zhang. Morse set classification and hierarchical refinement using Conley index. *IEEE Transactions on Visualization and Computer Graphics*, 18(5):767–782, 2012.

[5] G. Chen, K. Mischaikow, R. Laramee, P. Pilarczyk, and E. Zhang. Vector field editing and periodic orbit extraction using Morse decomposition. *IEEE Transactions on Visualization and Computer Graphics*, 13(4):769–785, 2007.

[6] G. Chen, K. Mischaikow, R. Laramee, and E. Zhang. Efficient Morse decompositions of vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 14(4):848–862, 2008.

[7] C. Conley. *Isolated invariant sets and the Morse index*. American Mathematical Society, Providence, RI, USA, 1978.

[8] W. de Leeuw and R. van Liere. Collapsing flow topology using area metrics. In *Proceedings of IEEE Visualization Conference*, pages 349–354, 1999.

[9] W. de Leeuw and R. van Liere. Visualization of global flow structures using multiple levels of topology. *Data Visualization*, pages 45–52, 1999.

[10] W. de Leeuw and R. van Liere. Multi-level topology for flow visualization. *Computers and Graphics*, 24(3):325–331, 2000.

[11] T. Delmarcelle and L. Hesselink. The topology of symmetric, second-order tensor fields. *Proceedings of IEEE Visualization Conference*, pages 140–147, 1994.

[12] T. Dey and R. Wenger. Stability of critical points with interval persistence. *Discrete Computational Geometry*, 38:479–512, 2007.

[13] H. Edelsbrunner, J. Harer, and A. Zomorodian. Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. *Discrete Computational Geometry*, 30:87–107, 2003.

[14] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete Computational Geometry*, 28:511–533, 2002.

[15] H. Edelsbrunner, D. Morozov, and A. Patel. The stability of the apparent contour of an orientable 2-manifold. In *Topological Methods in Data Analysis and Visualization*, pages 27–41. Springer-Verlag, 2010.

[16] H. Edelsbrunner, D. Morozov, and A. Patel. Quantifying transversality by measuring the robustness of intersections. *Foundations of Computational Mathematics*, 11:345–361, 2011.

[17] M. Floater. Mean value coordinates. *Computer Aided Geometric Design*, 20:19–27, 2003.

[18] A. Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[19] E. Hawkes, R. Sankaran, P. Pébay, and J. Chen. Direct numerical simulation of ignition front propagation in a constant volume with temperature inhomogeneities: II. Parametric study. *Combustion and Flame*, 145:145–159, 2006.

[20] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *IEEE Computer*, 22(8):27–36, 1989.

[21] T. Klein and T. Ertl. Scale-space tracking of critical points in 3D vector fields. *Topology-based Methods in Visualization Mathematics and Visualization*, pages 35–49, 2007.

[22] R. Laramee, H. Hauser, L. Zhao, and F. Post. Topology based flow visualization: the state of the art. In *Topology-Based Methods in Visualization (Proceedings of Topo-in-Vis 2005)*, Mathematics and Visualization, pages 1–19. Springer, 2007.

[23] M. Maltrud, F. Bryan, and S. Peacock. Boundary impulse response functions in a century-long eddying global ocean simulation. *Environmental Fluid Mechanics*, 10:275–295, 2010.

[24] K. Polthier and E. Preuß. Identifying vector fields singularities using a discrete hodge decomposition. In *Visualization and Mathematics III*, pages 112–134. Springer-Verlag, 2003.

[25] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 1992.

[26] J. Reininghaus, J. Kasten, T. Weinkauf, and I. Hotz. Efficient computation of combinatorial feature flow fields. *IEEE Transactions on Visualization and Computer Graphics*, 2011.

[27] J. Reininghaus, N. Kotava, D. Guenther, J. Kasten, H. Hagen, and I. Hotz. A scale space based persistence measure for critical points in 2d scalar fields. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2045–2052, 2011.

[28] J. Reininghaus, C. Lowen, and I. Hotz. Fast combinatorial vector field topology. *IEEE Transactions on Visualization and Computer Graphics*, 17(10):1433–1443, 2011.

[29] H. Theisel, C. Rössl, and H.-P. Seidel. Combining topological simplification and topology preserving compression for 2D vector fields. In *Pacific Graphics*, pages 419–423, 2003.

[30] H. Theisel, C. Rössl, and H.-P. Seidel. Compression of 2D vector fields under guaranteed topology preservation. *Computer Graphics Forum*, 22(3):333–342, 2003.

[31] Y. Tong, S. Lombeyda, A. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM Transactions on Graphics*, 22:445–452, 2003.

[32] X. Tricoche, G. Scheuermann, and H. Hagen. A topology simplification method for 2D vector fields. In *Proceedings of IEEE Visualization Conference*, pages 359–366, 2000.

[33] X. Tricoche, G. Scheuermann, and H. Hagen. Continuous topology simplification of planar vector fields. *Proceedings of IEEE Visualization Conference*, pages 159–166, 2001.

[34] X. Tricoche, G. Scheuermann, and H. Hagen. Topology-based visualization of time-dependent 2D vector fields. In *Proceedings Joint Eurographics-IEEE TCVG Symposium on Visualization Proceedings*, pages 117–126, 2001.

[35] X. Tricoche, T. Wischgoll, G. Scheuermann, and H. Hagen. Topology tracking for the visualization of time-dependent two-dimensional flows. *Computers & Graphics*, 26:249–257, 2002.

[36] B. Wang, P. Rosen, P. Skraba, H. Bhatia, and V. Pascucci. Visualizing robustness of critical points for 2d time-varying vector fields. *EuroVis: Eurographics Conference on Visualization (accepted)*, 2013.

[37] R. Westermann, C. Johnson, and T. Ertl. A level-set method for flow visualization. *Proceedings of IEEE Visualization Conference*, pages 147–154, 2000.

[38] E. Zhang, K. Mischaikow, and G. Turk. Vector field design on surfaces. *ACM Transactions on Graphics*, 25:1294–1326, 2006.