# Critical Point Cancellation in 3D Vector Fields: Robustness and Discussion

Primoz Skraba, Paul Rosen, Bei Wang, Guoning Chen, Harsh Bhatia and Valerio Pascucci

**Abstract**—Vector field topology has been successfully applied to represent the structure of steady vector fields. Critical points, one of the essential components of vector field topology, play an important role in describing the complexity of the extracted structure. Simplifying vector fields via critical point cancellation has practical merit for interpreting the behaviors of complex vector fields such as turbulence. However, there does not exist an effective technique that allows direct cancellation of critical points in 3D. This work fills this gap and introduces the *first* framework to directly cancel pairs or groups of 3D critical points in a hierarchical manner with guaranteed minimum amount of perturbation based on their robustness, a quantitative measure of their stability. In addition, our framework does not require the extraction of the entire 3D topology, which contains nontrivial separation structure, and thus is computationally effective. Furthermore, our algorithm can remove critical points in any subregions of the domain whose degree is zero and handle complex boundary configurations, making it capable of addressing challenging scenarios that may not be resolved otherwise. We apply our methods to synthetic and simulation datasets to demonstrate its effectiveness.

**Index Terms**—Flow visualization, Vector field simplification, Robustness, Computational topology

## 1 Introduction

Complex and often large-scale vector fields arise from a vast variety of scientific and engineering applications, including climate study, combustion dynamics, seismology, medicine, automobile and aircraft design. Topological methods have been employed extensively to extract features, such as critical points and separatrices (i.e., special streamlines or stream surfaces starting from the saddle points), for the purpose of vector field visualization [1], compression [2], design [3], [4] and simplification [5].

Applications requiring the study of turbulent flow in particular, may generate vector fields with a large number of critical points, leading to a visually cluttered representation that hinders intuitive interpretation of the flow behavior. A popular solution to address such a challenge is to simplify the flow by systematically reducing the number of critical points in the obtained topological representations. Such topology-based simplification schemes typically cancel pairs of critical points that are directly connected by separatrices in order of their importance based on certain geometric proximities (e.g., distance or area). While this strategy may work well for 2D vector fields [4], [5], [6], it is not straightforward to extend to 3D vector fields due to the increasing complexity of 3D topology. In addition, the full 3D vector field topology can potentially be expensive to extract due to the increased dimensionality of the separatrices as well as numerical instabilities [3], [7], [8], making it less practical for large-scale datasets. Furthermore, such simplification typically does not take into account the influence of flow magnitude, an important physical property of the flow.

In general, we believe that there has been little work done towards topological simplification of 3D vector fields, especially the development of techniques that do not depend upon the computation of the topological skeleton. In fact, the only work that we found in literature is based on the extraction and visualization of high-order critical points [8]. Simplification is achieved by looking at the behavior of the flow on a bounding surface surrounding a cluster of first-order critical points. A simplified representation of the 3D vector field topology is then obtained by merging lower-order critical points into higher-order ones. It has been shown that the subsequent vector field representation is substantially simplified accordingly after this merging process. Fundamentally different from their approach, we propose the *first* framework that *directly cancels* pairs of 3D critical points with *guaranteed minimum amount of perturbation* for 3D vector fields. Importantly, it does not require computing the entire topology of the vector field. Although the minimum amount of perturbation may indicate a small change to the vector field behavior in the vicinity of the critical points, it prevents the introduction of misleading information. Therefore, we believe that the reduced number of critical points will lead to a simplified topological representation of 3D flows. In particular, our contributions are as follows, (see Figure 1 for an illustrative example):

- We propose a novel, scalable and hierarchical simplification strategy for 3D vector fields, where sets of critical points are canceled based on a measure of their stability, quantified by the topological notion of robustness. The resulting vector fields are smooth, and with bounded perturbations.
- We generalize our algorithm to remove critical points in any connected region of the vector field (even the ones with non-trivial topology), as long as it has zero degree. It does not require special boundary configurations.


- *P. Skraba is with Jozef Stefan Institute, Slovenia. E-mail: primoz.skraba@ijs.si*
- *P. Rosen is with University of South Florida. E-mail: prosen@usf.edu*
- *B. Wang and V. Pascucci are with Scientific Computing and Imaging Institute, University of Utah. E-mails: {beiwang, pascucci}@sci.utah.edu*
- *G. Chen is with University of Houston. E-mail: chengu@cs.uh.edu*
- *H. Bhatia is with Lawrence Livermore National Laboratory. E-mail: bhatia4@llnl.gov*

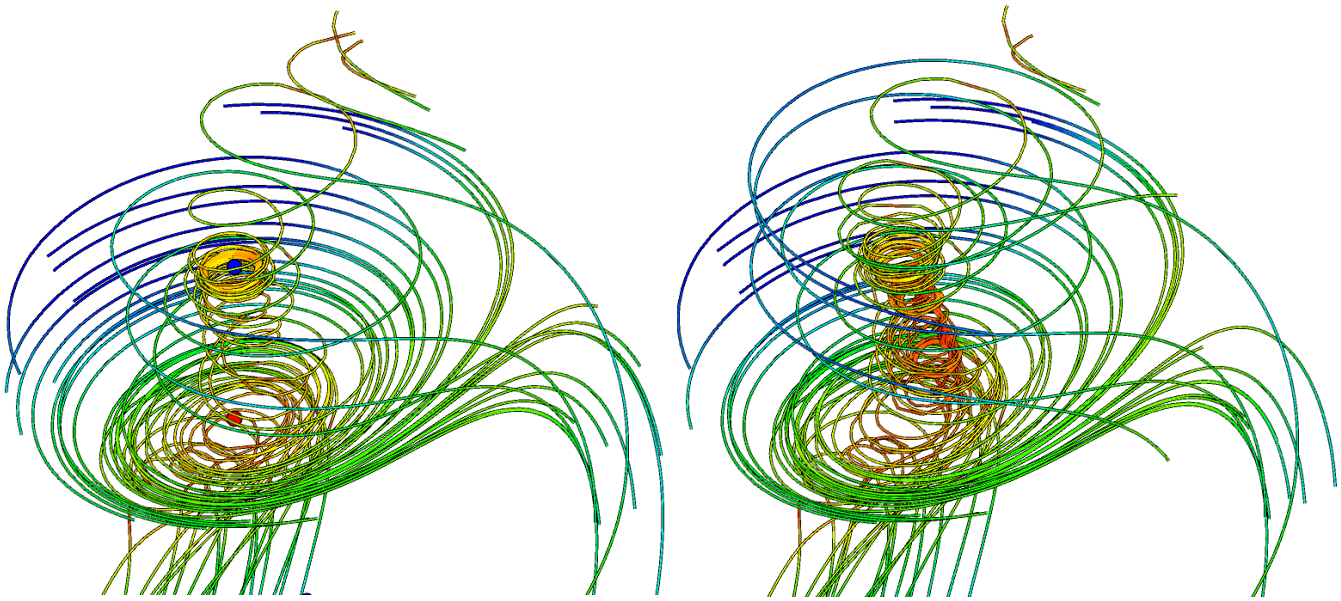

Fig. 1. A synthetic example that illustrates the complexity of 3D simplification and demonstrates our robustness-based simplification of critical points. For a pair of critical points residing within a region near a vortex core (as illustrated by the highly rotating streamlines), we showcase the global flow behavior before (left) and after (right) simplification. Before simplification (left), the identification of this vortex core may return a number of disconnected curves due to the existence of a pair of critical points (sinks are red, saddles are blue). Cancellation of these critical points (right) eliminates their interferences in the behavior of the vortex core. Namely, the vortex core exhibits higher continuity, which is highlighted by a bundle of red streamlines passing through the center of the region (i.e., without stopping at the original critical points). In this case, the simplification results in local rather than global changes to the flow behavior, as indicated by streamline modifications in close proximity of the critical points, and such local changes are crucial to the study of the spiral region and the identification of vortex core. See supplementary video for details.

## 2 Related Work

**Vector field simplification.** Vector field simplification can be accomplished via topology-based or non-topology-based techniques. Non-topology-based techniques [9], [10], [11], such as vector field clustering or segmentation, do not explicitly simplify the flow structure. Therefore, we concentrate on only the former.

Vector field topology has been applied successfully to the 2D vector field simplification problem [4], [6], [12], [13] with the aid of the notion of *topological skeleton* [5], [14]. Recently, Morse decomposition and Morse Connection Graph (MCG) have been introduced to the visualization community to study the topology of vector fields [15], [16]. This topological representation (i.e., MCG) can facilitate the construction of a hierarchy of the flow structure by either a refinement process [17] or the merging of neighboring Morse sets [18]; where the sizes of the Morse sets are used to determine the ordering of simplification or refinement.

Despite the extensive research on 2D vector field simplification based on topology, there exists little work on the topology-based simplification in 3D. This is, at least in part, due to the complexity of the topology of general 3D vector fields [3], [19] which consists of not only 1D skeleton but also 2D separation surfaces. This increased complexity has made the extraction and visualization of 3D vector field topology challenging [20], [21]. Theisel et al. introduced the saddle-connector to reduce the occlusion issue in the visualization of 3D topology [7]. Weinkauf et al. [3], [8] introduced a technique to visualize high-order critical points. This is achieved via the F-classification of a derived auxiliary tangential vector field defined on a closed surface surrounding each critical point. Iconic visualization can be produced using the minimal skeleton of this derived vector field. Simplification of a 3D vector field can be achieved by placing the closed surface described earlier around a group of first-order critical points. This is fundamentally different from our approach. While their method achieves a simplified representation by *merging* lower-order critical points into higher-order ones, our approach *cancels* pairs of first-order critical points.

**Robustness.** To introduce hierarchical simplification of vector fields, one would rank the critical points by measures of relevance or importance. Kasten et al. [22] ranked features based on their lifetime in the time-varying setting and treat long-lived ones as being significant. Reininghaus et al. [23] proposed a persistence-like importance measure for critical points, which discriminates between stable and unstable features of the vector field. Klein et al. [24] tracked vector field critical points over multiple spatial scales to assess their importance to the overall behavior of the flow field. In this paper, we use the topological notion of *robustness*, a relative of *persistence* [25], which was introduced through the algebraic concept of well diagrams in [26], [27], [28], to quantify the stability of critical points with respect to perturbations, which is crucial in assessing their significance. It has been shown to be useful for the analysis and visualization of both stationary and time-varying 2D vector fields [29], [30]. It also leads to interesting theoretical results in feature tracking by relating critical points correspondences with their stability [31]. Recent work in [32] proposed a 2D vector field simplification strategy based on robustness, where critical points are canceled according to a quantitative measure of their stability. Such a strategy provides a complementary view to topological-skeleton-based simplification and handles more general boundary configurations. In this paper, we provide a 3D version of such a strategy, which fills the gaps of direct cancellation of pair of critical points in 3D vector fields. Compared to its 2D counterpart, our 3D robustness-based simplification is technically much more sophisticated and has, potentially, an even bigger impact on enriching the research field

of 3D vector field simplification.

## 3 Technical Background

We provide relevant background in degree theory and robustness. We give minimal algebraic definitions and offer a high-level intuition whenever possible.

Our work relies on a corollary of the Poincaré-Hopf theorem, which states that a compact, connected, oriented manifold $\mathbb{M}$ possesses a nowhere-vanishing vector field if and only if its Euler characteristic is zero ( [33], page 146). This corollary implies that given a disk $D \subseteq \mathbb{M} \subset \mathbb{R}^m$ that contains multiple critical points of $f$, one could obtain a (homotopic) simplification of $f$ by removing all the critical points in $D$ while leaving $f$ constant outside of $D$.

**Degrees and indices.** Consider $f : \mathbb{M} \to \mathbb{R}^m$ to be a vector field on a manifold $\mathbb{M}$ (where $m = 3$ in our context). Suppose $x$ is an isolated zero (i.e. critical point) of $f$, that is, let $|f(x)| = 0$. Fix the local coordinates near $x$ and pick a closed disk $D$ which encloses $x$ in its interior and contains no other critical points. Then the *index* of $x$, or equivalently the *local degree* of $f$ at $x$, denoted as $\deg(f|_x)$, is the degree of the map $u : \partial D \to \mathbb{S}^{m-1}$ that maps the boundary of $D$ to the $(m-1)$-sphere, given by $u(z) = f(z)/|f(z)|$ ($u$ is sometimes referred to as the *Gauss* map). If $D$ is a disk that contains multiple critical points $\{x_1, x_2, ..., x_n\}$ of $f$, then the degree of $f$ restricted to $\partial D$ is the sum of the indices/degrees of $f$ at the $x_i$, $\deg(f|_{\partial D}) = \sum_{i=1}^{n} \deg(f|_{x_i})$. For notational convenience, when $f$ is fixed, we abuse notation by defining $\deg(D) := \deg(f|_{\partial D})$ and $\deg(x_i) := \deg(f|_{x_i})$, and refer to them as degrees of $D$ and $x_i$ respectively.

We consider isolated, first-order critical points in 3D: sinks, sources, repelling saddles and attracting saddles. Their indices (degrees) are +1, -1, +1 and -1 respectively.

**Merge tree.** Given an $m$-dimensional continuous vector field $f : \mathbb{M} \subset \mathbb{R}^m \to \mathbb{R}^m$, we define its corresponding magnitude function $f_0(x) = ||f(x)||_2$. We use $\mathbb{F}_r = f_0^{-1}(-\infty, r]$ to denote its *sublevel set* for some $r \geq 0$. $\mathbb{F}_0$ is the set of critical points of $f$. To compute robustness of critical points in $f$, we construct a merge tree of $f_0$ that tracks the (connected) components of $\mathbb{F}_r$ (together with their degree information) as they appear and merge by increasing $r$ from 0. A leaf node represents the creation of a component at a local minima of $f_0$ and an internal node represents the merging of components, see [29], [34] for algorithmic details. To illustrate the construction, we show a 2D example adapted from [29] in Fig. 2. Once the critical points and degrees are computed, the construction of the merge tree is independent of the dimension, since it uses only 0-connectivity (connected components) of the sublevel sets.

**Robustness.** The robustness of a critical point is the height of its lowest zero degree ancestor in the merge tree [29], [35]. For example in Fig. 2, by definition, the critical points $x_1$ and $x_2$ have robustness $r_1$, $x_3$ and $x_4$ has robustness $r_3$. Such a topological notion quantifies the stability of a critical point with respect to perturbations of the vector fields. The robustness of a critical point is the minimum amount of perturbation required to cancel it. Its technical properties are explicitly stated in [29], which we restate here for completeness. A continuous mapping $g$ is an *r-perturbation* of $f$, if $\mathrm{d}(f,g) \leq r$, where $\mathrm{d}(f,g) = \sup_{x \in \mathbb{R}^2} ||f(x) - g(x)||_2$. Suppose a critical point $x$ of $f$ has robustness $r$, then:

***Lemma 3.1 (Critical Point Cancellation [29]).*** Let $D$ 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 $h^{-1}(0) \cap D = \emptyset$ and $h = f$ except possibly within the interior of $D$.

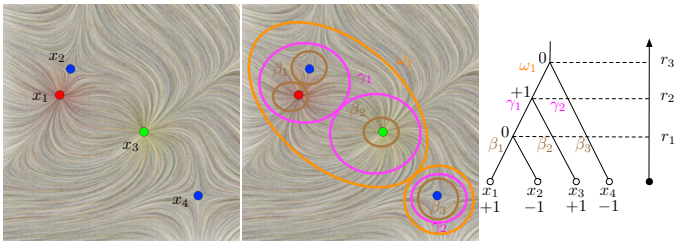


Fig. 2. Figure from [29] showing the merge tree for a 2D vector field example. The robustness computation and merge tree construction are the same in all dimensions. Suppose the vector field is continuous, where sinks are red, sources are green, and saddles are blue. From left to right: vector fields $f$, relations among components of $\mathbb{F}_r$, and the augmented merge tree. $f$ contains four critical points, a sink $x_1$, a source $x_3$, and two saddles $x_2$ and $x_4$. We use $\beta$, $\gamma$, $\omega$, etc. to represent components of the sublevel sets.

***Lemma 3.2 (Degree Preservation [29]).*** Let $D$ be the connected component of $\mathbb{F}_{r-\delta}$ containing $x$, for some $0 < \delta < r$. For any $\varepsilon$-perturbation $h$ of $f$ ($\varepsilon \leq r - \delta$), $\deg(f|_{\partial D}) = \deg(h|_{\partial D})$. If $D$ contains only one critical point $x$, $\deg(h|_{\partial D}) = \deg(f|_x)$.

Intuitively, if a critical point $x$ has robustness $r$, then it can be canceled with a $(r+\delta)$-perturbation, but not with a $(r-\delta)$-perturbation, for $\delta > 0$ arbitrarily small. By construction, our simplification strategy perturbs the vector field by $r + \delta$, introducing the smallest possible (i.e. optimal) point-wise perturbation.

## 4 Robustness-Driven Simplification Algorithms

Suppose we are given a 3D vector field with the critical points identified along with their local degree information. During robustness-driven simplification, we first compute the robustness values of each critical point. For critical points that share the same robustness value of $r$, we compute the corresponding component of the sublevel set with minimum area, $D \subseteq \mathbb{F}_r$, that encloses them. Due to the inherent properties of robustness, by construction $\deg(D) := \deg(f|_{\partial D}) = 0$. We then apply our simplification strategy to simplify $f$ in $D$ while leaving $f$ intact outside of $D$. It is important to point out, that although the vector field modification is local it can have global effects on the flow behavior (Fig. 1).

### 4.1 Preliminary

**Preprocessing: robust critical detection.** For simplicial meshes, the simplest way to detect critical points is solving a linear system numerically for each tetrahedron to identify if there exists a zero in its function space. However, the critical points detected this way suffer greatly from numerical instabilities, often creating false negatives and/or false positives, especially when the induced linear system is poorly conditioned. Instead, we employ a numerically robust approach [36], which uses symbolic perturbations [37] to eliminate numerical instabilities.

**Preprocessing: degree and robustness computation.** To compute robustness, we must also compute the local degrees (along with their relative degree orientations) of the detected critical points. For simplicity, we refer to the degree of a critical point $p$ in a tetrahedron $t$ as the degree of $t$. Suppose there are only first-order critical point in the vector field, a tetrahedron $t$ (with

a critical point $p$ in its interior) may only be assigned a degree of $+1$ or $-1$, whereas regions larger than a tetrahedron may have higher degrees.

Suppose the tetrahedron $t$ is formed by four vertices $\{a,b,c,d\}$, with edges denoted by $ab$, $ac$, $ad$, etc. The *orientation* of $t(a,b,c,d)$ is defined as $\text{sign}(ab \cdot (ac \times ad))$, this is the mixed product or *determinant* of $|(ab)(ac)(ad)|$ and $b$ sees the triangle $acd$ as counter-clockwise. To compute the degree orientation of a given tetrahedron $t$, we examine the orientation of $t$ in $\mathbb{R}^3$ as well as the orientation of $f(t)$. If the orientation of $t$ and $f(t)$ agrees, then the degree of $t$ is $+1$; otherwise it is $-1$. There are two possible scenarios: (a) if the origin $O$ (i.e., the image of critical point $p$ in $t$ under $f$, $f(p) = O$) is contained in the interior of $f(t)$; and (b) if the origin $O$ lies on (or near) the boundary of $f(t)$ or $f(t)$ is degenerate.

Scenario (a) is the generic situation. We compute the sign of the determinants of $t$ and $f(t)$ directly. Since we use regular tetrahedra $t$ in the domain, this ensures that the sign of its determinant can be correctly determined. Likewise, since the origin is in the interior of $f(t)$, the sign of its determinant can also be correctly determined.

In scenario (b), the above approach does not yield reliable results. Thus, we compute the degree orientation of a larger neighborhood $t'$ such that it encloses the given tetrahedron $t$. Let $t'$ be the collection of tetrahedra that are adjacent to $t$, $\partial(t')$ be its boundary and $f(\partial(t)')$ be the image of the boundary. Suppose $t'$ contains a single critical point in its interior, denoted as $p$, where $f(p) = O$ in the image space. We choose a random vector $r$ and detect the set of triangles $T$ in $f(\partial(t)')$ that intersect $r$. Although the choice of random vector $r$ is arbitrary in theory, we repeat the procedure with several random vectors in case of potential degeneracies.

We first determine the relative orientation of each such triangle. Suppose a triangle in $T$ corresponds to a triangle $abc$ in $t'$, we form the first tetrahedron in the domain as $t_1(a,b,c,p)$ and the second one in the image space as $t_2(f(a),f(b),f(c),O)$. We compare the orientation of $t_1$ and $t_2$ – if their signs agree, we say the contribution of the triangle $f(a)f(b)f(c)$ is $+1$, otherwise it is $-1$. This process is illustrated in Fig. 3 (left). To compute the degree, we sum the contributions of all the triangles in $T$. Such an idea corresponds to a high-dimensional analogue of computing winding number of a closed curve in the plane around a given point, as illustrated in Fig. 3 (right), where the winding number is computed by counting the number of up-crossings and down-crossings. The local degree computation is implemented using the CGAL library [38] with exact constructions and predicates.

For both of these techniques, we assume a piecewise linear (PL) interpolation. The computation of the determinant (scenario (a)) to detect the sign of the relative orientation requires the convexity of the tetrahedra. The alternate approach (scenario (b)) similarly requires the convexity of the boundary triangles. A more general interpolation scheme would rely on subdividing the triangles and applying PL interpolation to each of the pieces. In particular, this would only require the subdivision of triangles which intersect a chosen vector $r$. Note that this subdivision approach can detect higher order singularities (e.g. with absolute degrees larger than 1). Furthermore, although a simple intersection test is sufficient for our purposes, there is room for improvement. We could use more advanced tests (e.g. [39]) to robustly detect intersections between the vector and triangles in image space.

Finally, we compute robustness associated with each critical point by following the merge tree algorithm described in Section 3 and detailed in [29]. Robustness computation is implemented using C++ and is roughly linear in the number of tetrahedra.

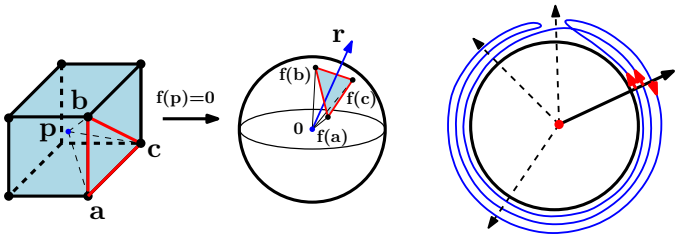


Fig. 3. Left: mapping between tetrahedra $(a,b,c,p)$ and $(f(a),f(b),f(c),O)$. Right: computing winding number of a curve, where winding# = #up-crossings − #down-crossings $= 2-1=1$.

**Image space.** Given a 3D vector field restricted to a degree-zero component $D \subset \mathbb{F}_r$, $f : D \to \mathbb{R}^3$, we define the *image space* of $D$, $\text{im}(D)$, by mapping each point $p \in D$ to its vector coordinates $f(p)$ in $\mathbb{R}^3$. *Intuitively*, this is similar to the concept of a Gauss map except that the vectors are not normalized. Since $D \subseteq \mathbb{F}_r$, it follows that $\text{im}(D)$ is contained within a 3-ball of radius $r$, denoted as $\mathbb{S}^3_r$, whose boundary is a 2-sphere, denoted as $\mathbb{S}^2_r$. Similar to the 2D case [32], we consider the boundary of $\text{im}(D)$ *uncovered*, if $\text{im}(\partial D) \subset \mathbb{S}^2_r$; otherwise, as *covered*. These concepts naturally extend to the piece-wise linear setting where we consider a vector field $f$ restricted to a triangulation $K$ of $D$, $f : K \to \mathbb{R}^3$ where the support of $K$ equals $D$. The above construction then maps a triangulation $K$ of $D$ to vertices, edges, triangles and tetrahedra in $\text{im}(D)$. An important feature of this approach is that it does not require the sublevel set component to be simply connected. Likewise, this implies that the $\partial D$ may not be simply connected or may even consist of multiple components.

### 4.2 Algorithm Details

**Algorithm overview.** Given a zero-degree connected component $D \subseteq \mathbb{F}_r$, our simplification strategy consists of three steps.

- **(a)** We perform Laplacian smoothing on $D$ (i.e. the **Smoothing** operation). If the resulting vector field does not contain critical points, return $D_1 = \textbf{Smoothing}(D)$.
- **(b)** We randomly sample points uniformly in $\mathbb{S}^2_r$. If among all the sampled points, there exists a point $p$ such that it belongs to the uncovered region of $\mathbb{S}^2_r$, that is $p \in (\mathbb{S}^2_r - \text{im}(\partial D))$, then with respect to $p$, we: (i) deform the vector field in its image space $\text{im}(D)$ to remove critical points in $D$ (i.e. **Cut** operation); (ii) set $D_1 = \textbf{Cut}(D)$, and return $D_2 = \textbf{Smoothing}(D_1)$.
- **(c)** If no such point $p$ has been found with sufficient samples, then we assume the boundary of $\text{im}(D)$ is covered. In this case, we perform **Unwrap**. We modify the vector field in its image space $\text{im}(D)$ so part of its boundary becomes uncovered (i.e. **Unwrap** operation). We set $D_1 = \textbf{Unwrap}(D)$, $D_2 = \textbf{Cut}(D_1)$, and return $D_3 = \textbf{Smoothing}(D_2)$.

The final step **(d)** is the **Restore** operation where we set the boundary to its original value. We now describe some key operations in detail.

**Smoothing operation.** Given a vector field $f$ and a region $D$ to be simplified, a modified vector field $\overline{f}$ inside $D$ can be found by solving a constrained optimization problem, referred to as *Laplacian smoothing*. Specifically, a vector-valued discrete Laplacian equation is solved over $D$ in the domain (e.g., a triangular

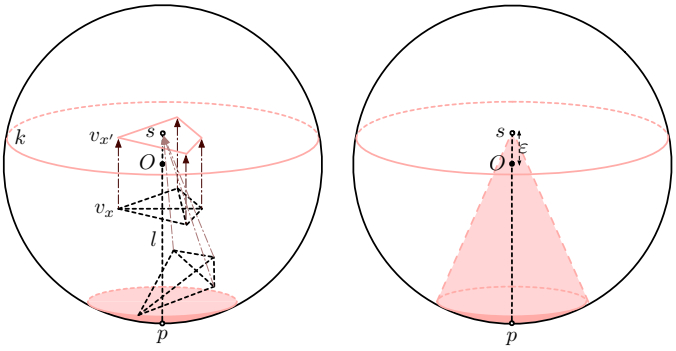


Fig. 4. **Cut** operation. Left: the projection of tetrahedra that intersect $l$, the shaded region represents the uncovered surface of $\mathbb{S}^3_r$. Right: after **Cut**, the shaded region corresponds to the empty wedge (void) containing the origin. The resulting vector field is critical point free.

mesh) where the vector values at the boundary vertices of $D$ 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 $v_j$ are the adjacent vertices that are either in the interior or on the boundary of $D$. The weights $\omega_{ij}$ are usually determined using Floater's mean-value coordinates [40]. This is a sparse linear system, which can be solved using a conjugate gradient method [41]. 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 linear system solver and the spatial discretization.

**Cut operation.** Recall that the image space $\mathrm{im}(D)$ of a component $D \subseteq \mathbb{F}_r$ by construction belongs to a 3-ball $\mathbb{S}^3_r$. If $D$ contains critical points, this implies that the center $O$ of $\mathbb{S}^3_r$ (i.e. the origin) is part of $\mathrm{im}(D)$. The **Cut** operation then deforms $\mathrm{im}(D)$ such that there is a small void surrounding the origin, which means that the resulting vector fields restricted to $D$ does not contain any critical points. The key idea is to perform such a deformation via a so-called *cut point* $p$ on the uncovered part of $\mathbb{S}^2_r$. Although a deterministic algorithm exists to locate a valid cut point via stereographic projection, in practice, uniform random sampling on $\mathbb{S}^2_r$ works well in detecting cut points quickly and accurately as well as being far simpler to implement. Suppose among all the sampled points, there exists a point $p \in \mathbb{S}^2_r - \mathrm{im}(\partial D)$.

As illustrated in Figure 4, to deform the vector field, we first define a line segment $l$ that connects the origin $O$ and the cut point $p$. We define a 2D plane $k$ that is orthogonal to $l$ but is $\varepsilon$ away from $O$. $O$ projects onto $k$ at a point $s \in k$. Next, we find all tetrahedra in the interior of $\mathrm{im}(D)$ that intersect with $l$ and project their boundary points onto $k$, whose projections form triangles in $k$, e.g. in the domain of the vector field $D$, vector at $x \in K$ is deformed from $v_x$ to $v_{x'}$ respectively. Third, we locate all tetrahedra that intersect $l$ and contain one or two boundary points in $\mathrm{im}(\partial D)$. For each such tetrahedron, we move its remaining boundary points to $s$. Since point $p$ is uncovered, there exists no triangle that intersects $l$ whose boundary points are all on the boundary of $\mathrm{im}(D)$. This operation creates an empty wedge around $O$ which ensures that there are no critical points in $D$ after the modification. By construction, the amount of perturbation is less than $r + \varepsilon$. This represents the bound on the amount of perturbation for simplification. Note that **Smoothing** operation may increase the amount of perturbation, but it is not strictly required for simplification.

In practice, the effect on image space through the cut operation can be seen in Fig. 5 (c-e) (please refer to Section 5 for details on the datasets). If the boundary is uncovered as in Figure 5 (b), the internal tetrahedra may be projected to uncover the origin and hence remove the critical points.

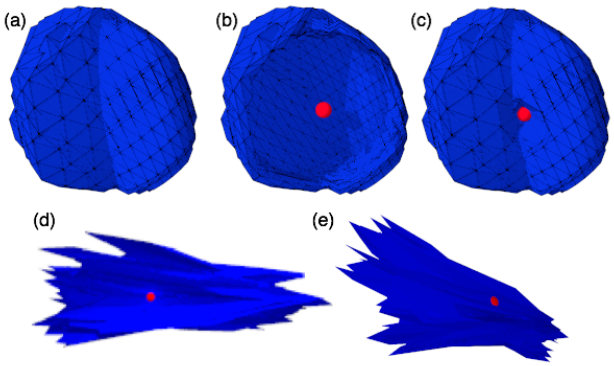


Fig. 5. Top: image space for Synthetic#1 before (a) and after (c) simplification via the cut operation. (b) shows only the boundary triangles, where the uncovered origin (red sphere) is visible. Since it is visible, the origin is uncovered and therefore there are no critical points remaining after simplification in the sublevel set. Bottom: in practice, the image space may be highly irregular both before and after simplification via the cut operation. For example, we show the image space for DeltaWing#1 (d) and #2 (e) after the simplification, where the origins (where critical points used to map to) are visible (therefore uncovered).

**Unwrap operation for covered image space.** The final scenario is that when the degree of $D$ is 0, however $\mathrm{im}(\partial D)$ covers $\mathbb{S}^2_r$. The **Cut** cannot be directly applied since its corresponding sampling procedure relies on $\mathrm{im}(\partial D)$ not covering the entire sphere in the image space. In this case, the boundary must first be unwrapped to reveal an uncovered region before the cutting can take place. In 2D, as shown in [29], this case happens rarely, occurring only in specially generated datasets. Surprisingly, this situation occurrs more readily in 3D, appearing in generic synthetic datasets as well as real-world ones. Intuitively it seems that it would be difficult to cover the entire sphere with a degree zero map. However, the increased complexity of the boundary of the component seems to imply the existence of such instances. In practice, it has occurred for robust pairs where the corresponding sublevel sets are quite large and complex (see Synthetic#2 in Section 5.4 for an example).

In this scenario, since no potential cut point can been found, the simplification would fail at the random sampling stage. While theoretically the procedure for simplification is dimension-independent, there are significant obstacles for the unwrapping step in 3D as opposed to 2D. In 2D, the boundary maps to the circle $\mathbb{S}^1$. Since $\mathbb{R}$ is the *universal cover* of $\mathbb{S}^1$, it is possible to unwrap the image of the boundary, using the parameterization by the angle (see [29] for details). In 3D, this is not possible, as the universal cover of $\mathbb{S}^2$ is again $\mathbb{S}^2$ itself. There exists no mapping into the Euclidean space $\mathbb{R}^2$, from which to determine the unwrap point. The obvious extension of the 2D case, using two angles for unwrapping unfortunately does not work. This describes a mapping into a torus (i.e. $\mathbb{T} = \mathbb{S}^1 \times \mathbb{S}^1$ ) rather than a sphere.

Our approach is an iterative smoothing procedure along a sphere, peeling back the boundary until a point is uncovered. Then, we proceed with the remaining steps, i.e. cutting, smoothing and restoring the boundary. The intuition behind this approach is that the Hopf degree theorem [33] states that any map $f : \mathbb{S}^n \to \mathbb{S}^n$ is homotopic to a constant map if and only if it has degree 0. The degree 0 condition therefore guarantees that we can continuously deform $\mathrm{im}(\partial D)$ to a single point. We do not deform $\mathrm{im}(\partial D)$ all the way to a point, but until we can find an uncovered region (therefore leading to a cut point).

The iterative procedure consists of two steps and is performed within the space $\mathrm{im}(\partial D)$ (i.e. the image of the boundary triangles in $D$). $\mathrm{im}(\partial D)$ is supposed to cover $\mathbb{S}^2_r$, the 2-sphere with radius $r$ (where $r$ is the robustness value, i.e. the largest vector magnitude in $D$). In other words, the underlying space of $\mathrm{im}(\partial D)$, $|\mathrm{im}(\partial D)| \subseteq \mathbb{S}^2_r$. The first step is similar to Laplacian smoothing of $\mathrm{im}(\partial D)$, in each iteration we set each vertex to the centroid of its neighbors. Since we take the Euclidean centroid, the vertex values move off of $\mathbb{S}^2_r$ over time. Therefore after each iteration, we renormalize the vertex values to project them back onto $\mathbb{S}^2_r$. To prevent the vertex values from moving too far off the surface of the sphere, we add a damping term. In each iteration, for each $v_i$,

1. $f(v_i) = (1-\alpha)f(v_i) + \alpha \sum_j \omega_{ij} f(v_j)$ where $0 < \alpha \leq 1$, $v_j$ are neighbors of $v_i$ and $\omega_{ij}$ are convex coefficients as in Laplacian smoothing.
2. $f(v_i) = \frac{r}{||f(v_i)||} f(v_i)$.

This is equivalent to Laplacian smoothing restricted to the sphere $\mathbb{S}^2_r$. In practice, $\mathrm{im}(\partial D)$ does not lie perfectly on a sphere. In some cases, where the vector field changes abruptly or if we are near the boundary of the domain, the magnitudes of vector field on the boundary vertices may be quite different (see Fig. 15 left). Therefore, we initially, and after each iteration, normalize/project the points on $\mathrm{im}(\partial D)$ to the sphere $\mathbb{S}^2_r$, see Fig. 6 for an illustration.

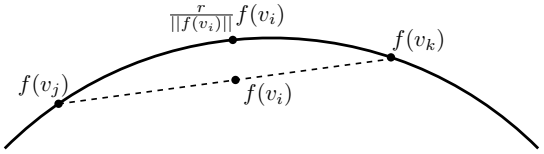


Fig. 6. The centroid of a set points on a sphere is not on the sphere itself. In this case $f(v_i)$ is the the centroid of its neighbors $f(v_j)$ and $f(v_k)$. Therefore after each iteration we perform normalization of image space to project the points back onto the sphere.

In our pipeline, the **Unwrap** operation is combined with the detection of the cut point to determine the termination condition for unwrapping. After every $k$ iterations, we perform a search for a cut point, terminating if successful. We give an example of such an unwrapping process in Section 5.4.

The method is also guaranteed to converge. This can be seen by considering the eigenvalues of the discrete averaging operator. In Euclidean space, it is known that since the eigenvalues are bounded by 1, repeated iterations converge. Although in our case we are on a sphere, the argument still holds. The damped version of the averaging operator still has its eigenvalues bounded by 1 and by taking a small enough step size (the parameter $\alpha$), it can be shown that the re-normalization step does not effect convergence (as these can be modeled by a correction term).

**Visualization.** Our visualization is built using Java and OpenGL and runs interactively on a variety of desktop and laptop platforms. The original vector field is first loaded into the system (including vectors, vertices, and tetrahedra), then a stencil is placed on top of only the vectors that are effected by the cancellation. Critical points are drawn as outlined spheres. They are colored by their indices, $+1$, red or $-1$, blue. The streamlines are rendered as outlined colored lines. Streamlines are seeded using the vertices of the sublevel set as the starting locations. Each streamline is then traced both forward and backward in time until one of the following conditions is satisfied: (a) hitting the boundary of the domain (for the study of global/external flow behavior, e.g. Fig. 1) or the boundary of the sublevel sets (for the study of local/interior flow behavior); (b) reaching a critical point; and (c) reaching the maximum number of integration steps. The vector magnitude is used to color streamlines by applying a rainbow color map. In some cases, too many streamlines are generated for a given sublevel set boundary, leading to occlusion. In this case, we enable the culling of the streamlines based upon a density measure that computes the average distance of a streamline to any other streamlines. The optional streamline tracing spheres (see accompanying video), are outlined spheres that are set to move at a constant speed in the forward time direction along the streamlines in order to convey the direction of flow.

## 5 Experiments

We experiment with three datasets: one synthetic dataset and two simulations from real-world experiments. The first *Synthetic* dataset is provided by the authors of [42]. The second one that we refer to as the *Delta Wing* dataset is courtesy of Markus Rütten, DLR Göttingen, Germany. It is an unsteady simulation of a delta wing configuration for the study of vortex breakdown. In the simulation, a sharp-edged prismatic delta wing moves at subsonic speed with the characteristic vortical systems above the wing, and increasing angle of attack eventually leads to vortex breakdown [43]. The final dataset is the *Lifted Flame* dataset. It is a subvolume from a direct numerical simulation of a turbulent lifted ethylene jet flame [44] which results in a compressible and highly turbulent flow. We describe our critical points cancellation with static images captured via our software. For a dynamic viewing of the simplification results in 3D, please refer to our supplemental video.

### 5.1 Synthetic Dataset

According to [42], this dataset contains two spirals, two saddles, and one source. The given mesh contains $132,651$ vertices and $750,000$ tetrahedra. Our critical point detection has resulted in ten first-order critical points. The discrepancy may be due to numerical issues. The augmented merge tree structure (together with robustness values and degree information) among these critical points is shown in Fig. 7 (left). We analyze the cancellation of two groups of critical points here. The first group consists of two critical points requiring only **Cut** operation. We refer to it as Synthetic#1, which has a robustness valued of 13.2. The second group labeled as Synthetic#2 consists of the remaining eight critical points with robustness valued at 257.1. They are to be simplified in a sublevel set that includes all critical points. This is also the only complex scenario that we encounter that requires **Unwrap** operation. We defer its discussion to Section 5.4.

For the pair of critical points in Synthetic#1, Fig. 8 illustrates the relative location of its enclosing sublevel set (highlighted by gray transparent surface) with respect to the entire domain from three different viewing angles. The two critical points (pointed by the arrows) have opposite degrees, i.e., degree $+1$ (red ball) and degree $-1$ (blue ball). As we increase the threshold of sublevel set, we see that the originally separated connected components that contain the two critical points eventually merge with each other forming an arch-like geometry that encloses the pair.

To investigate the local flow behavior of Synthetic#1 before and after simplification, we employ streamline culling and focus on the sublevel set that encloses the pair from the third viewpoint in Fig. 8. Their corresponding visualizations are provided in Fig. 9, where we highlight the location of the critical points before

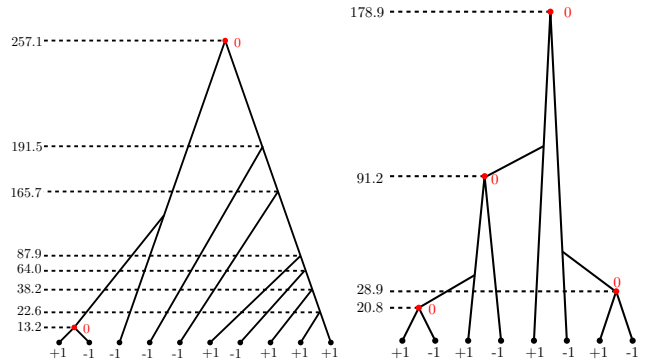


Fig. 7. Augmented merge trees for Synthetic (left) and Delta Wing (right) datasets.

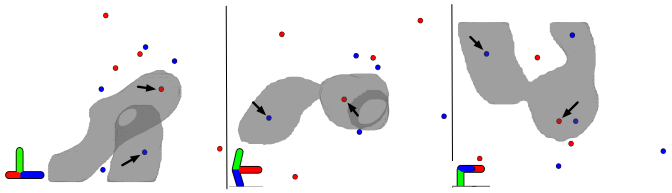

Fig. 8. Synthetic#1. Highlighted sublevel set from three different viewing angles. Critical points to be simplified are pointed by arrows.

simplification (left images). As illustrated in Fig. 9, Synthetic#1 is a typical 3D example of source-saddle cancellation, which is achieved by reversing the flow directions along the connection between them (similar to its 2D counterpart). After simplification, the flow near the original critical points has been slightly altered. This change is more obvious near the original source (red). Before simplification, the streamlines are all coming out of it, while after simplification, the streamlines traverse through it as illustrated in Fig. 9 (right). This indicates that the source has been removed and the corresponding flow region is critical-point-free. In the meantime, the flow away from the two canceled critical points remains mostly intact. This could be explained by the inherent characteristics of robustness-based simplification. The given critical points have relatively low robustness, therefore, the simplification requires only a small amount of perturbation (i.e., upper bounded by their robustness value), and results in vector field modification that is local to the sublevel set, which is desired for a topological simplification. The effect on the image space can also be seen in Fig. 5.

### 5.2 Delta Wing Dataset

The Delta Wing Dataset has been studied in previous work [43].The mesh contains $1,889,283$ vertices and $3,853,502$ tetrahedra. We detect eight saddles that form four pairs. The augmented merge tree structure (together with robustness values and degree information) among these critical points is shown in Fig. 7(right). We focus on three pairs of critical points for an in-depth investigation of the flow behavior before and after simplification. The cancellation of the last pair involves a very high robustness value with a sublevel set covering almost the entire domain, thus is omitted here. The three pairs are labeled as DeltaWing#1, #2 and #3, with robustness values 20.8, 28.9 and 91.2, respectively.

Fig. 10 illustrates the relative locations of their sublevel sets (highlighted by those gray transparent surfaces) with respect to the entire domain. The positioning of these sublevel sets specifically showcases the hierarchical simplification conducted at multiple levels of the merge tree. In particular, the sublevel set of DeltaWing#1 is entirely enclosed by the sublevel set of DeltaWing#3, forming a nesting configuration. This is also clear within the merge tree structure of Fig. 7(right).

The simplification results for DeltaWing#1 is shown in Fig. 11. Both DeltaWing#1 and #2 (omitted here) involve a pair of critical points located near a vortex core/spiral region, as illustrated by the circular patterns of the streamlines within the sublevel set. This is similar to the example in Fig. 1. After simplification, there exist no critical point in the interior of the spiral region, leading to a potentially continuous representation of the vortex core (not shown in here). This is desirable for the subsequent identification and visualization of vortices. We notice that the sublevel set modified appears to be long and relatively skinny. This is a challenging boundary configuration that our simplification algorithm can handle.

Fig. 12 demonstrates a case for DeltaWing#3 where more than two critical points could be canceled in a single simplification process. In this example, four critical points are enclosed within a single component of the sublevel set that has zero degree. In addition, two of these points form the previous critical point pair DeltaWing#2. There are two possible ways to cancel this group of critical points. On the one hand, we could cancel the pair #2 first before canceling the other two critical points in a hierarchical fashion. On the other hand, if we fix a simplification level, then these four critical points could potentially be cancelled together. In either case, our method can successfully replace the flow within the sublevel set region with the one free of critical points. We wish to point out that the latter group cancellation is particularly useful for the removal of a cluster of critical points with small robustness, which typically appear due to noise in the data or numerical instabilities. As illustrated in the zoomed views of Fig. 12, the flow before and after simplification has very minor changes, as illustrated by the similar patterns of their respective seeded streamlines. This provides another example to demonstrate that our robustness-based simplification requires only minimum amount of perturbation (bounded by the robustness) in order to remove critical points. It is worth mentioning that the sublevel set that encloses these four critical points has a non-trivial topology, that is, it contains a tunnel (see Fig. 13). This is an example where our algorithm could handle domains with arbitrary boundary configurations, even those that are not simply-connected. The only requirement on the zero degree domain is that it should be a *manifold* with boundary. This can always be achieved in the PL setting by appropriately thickening the original sublevel set.

### 5.3 Lifted Flame Dataset

We have extracted a small sub-volume of the Lifted Flame dataset for our experiment. The mesh contains $6,000,000$ vertices and $35,403,294$ tetrahedra. Since the flow field has a strong directional drift, we further subtract the mean field from the data. Such subtraction is essentially related to the Galilean invariance, see [45] for details. We analyze one pair of critical points in detail for demonstration purpose, it is labels as Flame#1, with a robustness value $1.4 \times 10^{-4}$. Fig. 14 shows the local flow behavior (restricted to the sublevel set) before and after simplification in the vicinity of the pair Flame#1. As can be seen, our robustness-based simplification successfully removes this pair of critical points with minimal amount of perturbation introduced to the vector field, i.e., the streamlines have similar patterns in general before and after simplification, except near the critical points.

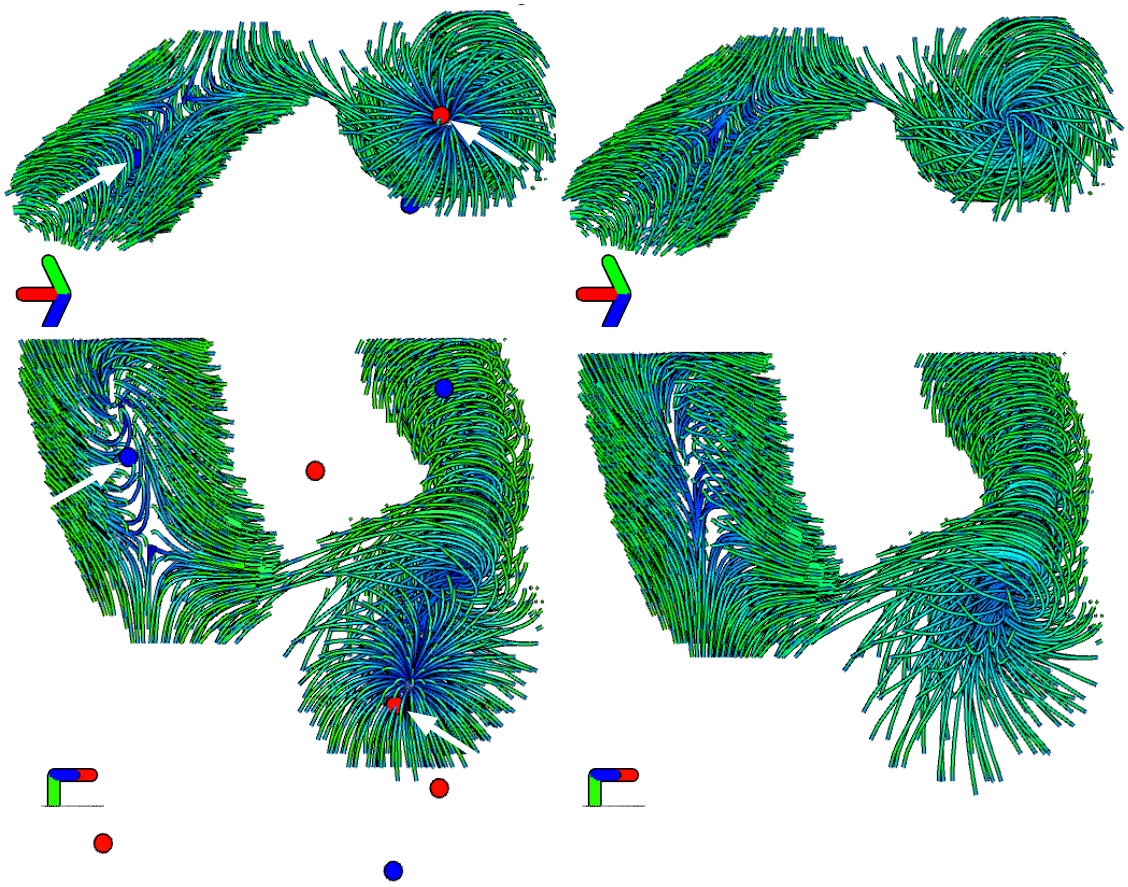

Fig. 9. Synthetic#1. Local view before (left) and after (right) simplification from the second (top) and third (bottom) viewing angle in Fig. 8. Critical points to be simplified are pointed by arrows.

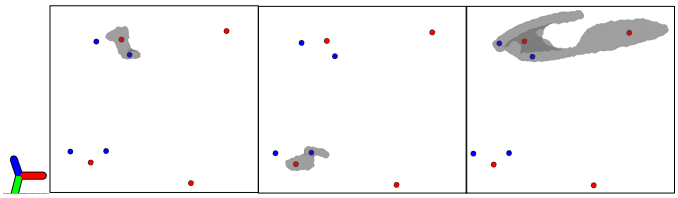

Fig. 10. Highlighted sublevel sets from three pairs of critical points to be simplified. From left to right, DeltaWing#1, #2 and #3 respectively.

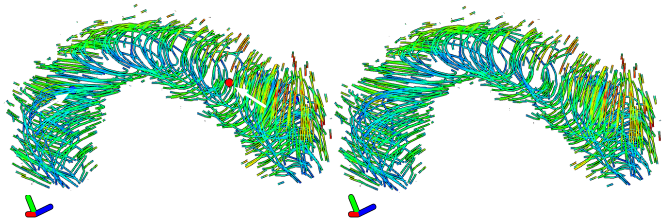

Fig. 11. DeltaWing#1 before (left) and after (right) simplification. Critical points to be simplified are pointed by arrows.

### 5.4 Discussion on Unwrap Scenario for Synthetic#2

The group of critical points in Synthetic#2 is the only complex scenario that we encounter that requires **Unwrap** operation. As illustrated by the merge tree structure in Fig. 7, the eight critical points have high robustness values (approximately four times of the robustness value of the Synthetic#1 pair), thus the sublevel set that encloses them almost covers the entire domain. Visualizing streamlines in the domain when all critical points have been removed does not give much useful information with respect to the simplification. Instead, in order to prove that our simplification algorithm involving the **Unwrap** operation is indeed correct, we illustrate the image spaces of the corresponding sublevel set before and after **Unwrap** in Fig. 15. On the left of Fig. 15, we see that the boundary covers the origin completely and that the magnitude of the vector field is very different across the boundary. This is due to the fact that such a component contains a large portion of the boundary of the domain. To apply our unwrapping algorithm, we normalize the vector field to the maximum of the magnitude (i.e. robustness value). This ensures that there are no points in the image of the component which are outside the sphere. The normalized vector field is shown in Fig. 15 (middle). At this stage, if we rotate the (re-normalized) image space, we see that there are no uncovered points. Furthermore, Fig. 15 (middle) shows that it is highly folded over itself with folds and cusps. After approximately 400 iterations, we obtain Fig. 15 (right), where we see that the unwrapping is successful (that is, there exists a region on the surface of the sphere that is uncovered). This implies that we could successfully locate a cut point within this uncovered region, and we proceed with the **Cut** and **Restore** operations.

## 6 Discussions

**Higher-order critical points.** Our simplification strategy can simplify isolated, first-order critical points in 3D vector fields. If we could detect higher-order critical points in 3D, e.g. using algorithms developed in [8], [21], in principle, our robustness-driven simplification strategy could be directly applied to canceling them. This is due to the fact that our algorithm only requires a

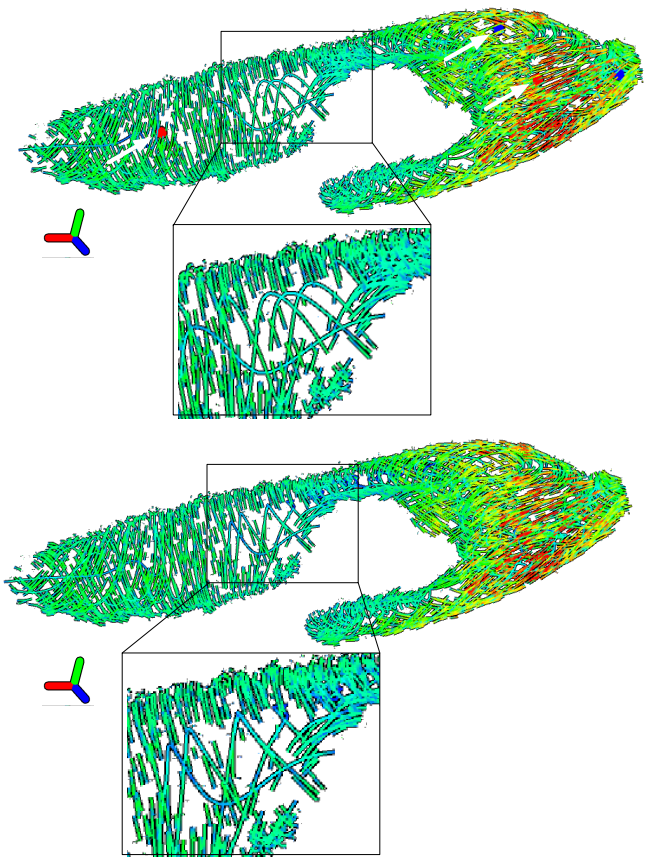

Fig. 12. DeltaWing#3. Cancellation of four critical points in a single setting, before (left) and after (right) simplification. The four critical points to be simplified are pointed by arrows.

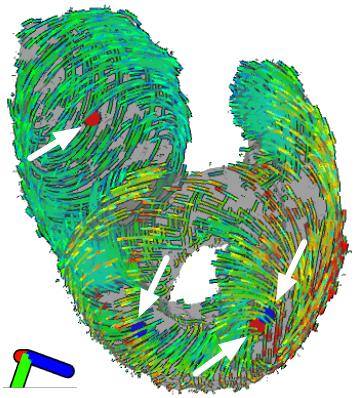

Fig. 13. DeltaWing#3. Sublevel set surrounding the four critical points (pointed by arrows) has non-trivial topology, e.g., it contains a tunnel.

zero-degree component in the sublevel sets and is oblivious to the degrees of individual critical points.

There may arise some situations where we may want to cancel particular sets of critical points. For example, in Synthetic dataset, we have a component of maximum degree of $+3$ at $38.2 < r < 64.0$ (Fig. 7 left). At $r = 64.0$, the component drops to a degree of 2. We may choose to cancel the $-1$ degree critical point with any of the three $+1$ degree critical points. Although there is no canonical choice of which critical point to cancel with, there certainly exists a connected component in the domain which contains the $-1$ degree point and one of the $+1$ degree points. Applying the algorithm on this subset would cancel this pair.

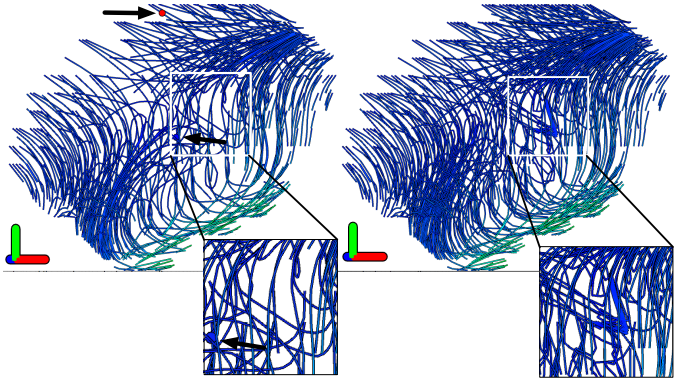

Fig. 14. Flame#1. Local flow behavior in the interior of the sublevel set before (left) and after (right) simplification.

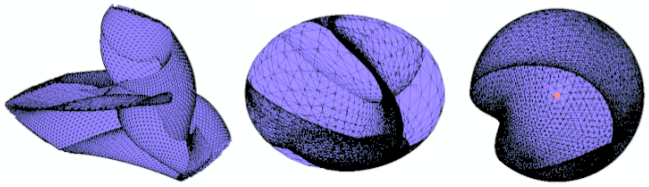

Fig. 15. The image space for Synthetic#2. On the left, we have the original image space, in the middle the normalized image space and on the right, the unwrapped image space. After **Unwrap**, we locate an uncovered region within the unwrapped image space on the right, and standard **Cut** operation is then performed followed by restoring the boundary.

**Vortex and vortex core structure.** Vortices are important flow structures that are not part of the vector field topology [46], [47], [48], [49]. However, there does not exist a unified definition of vortices. Existing vortex core detection techniques are also sensitive to small perturbation or noise in the data, leading to many small and disconnected segments as well as excessively complex vortex core structure. This prevents the identification of the salient behaviors of these vortices. Simplifying this vortex core structure by removing the smaller vortices can help alleviate this issue. However, there exists little work on the simplification of vortex core structure. Our robustness-based simplification can help achieve this goal to some extent. In the case of incompressible turbulent flow, we have seen a number of critical points near vortex cores (Fig. 1 and Fig. 11). The existence of these critical points partially contributes to the early termination of the tracing of vortex cores. Removing them with small amount of perturbation locally using our method can improve the continuity of the detected vortex cores and simplify the corresponding vortex core structure, while preserving the vortex behavior at the same time. Nonetheless, it will be interesting to extend the presented framework for the direct simplification of vortex core structure, which we plan to investigate in our future work.

## ACKNOWLEDGMENTS

PS was supported by TOPOSYS (FP7-ICT-318493) and PROASENSE (FP7-ICT-2013-10-612329). PR was supported by NSF ACI-1443046. BW and VP were supported by INL 00115847, DE-AC0705ID14517 and DOE NETL. GC was supported by NSF IIS-1352722.

## REFERENCES


[1] T. McLoughlin, R. S. Laramee, R. Peikert, F. H. Post, and M. Chen, "Over two decades of integration-based, geometric flow visualization," *Computer Graphics Forum*, vol. 29, no. 6, pp. 1807–1829, 2010.

[2] S. K. Lodha, J. C. Renteria, and K. M. Roskin, "Topology preserving compression of 2d vector fields," *IEEE Vis.*, pp. 343–350, 2000.

[3] T. Weinkauf, H. Theisel, H.-C. Hege, and H.-P. Seidel, "Topological construction and visualization of higher order 3d vector fields," *Comput. Graph. Forum*, vol. 23, no. 3, pp. 469–478, 2004.

[4] E. Zhang, K. Mischaikow, and G. Turk, "Vector field design on surfaces," *ACM Trans. on Graphics*, vol. 25, pp. 1294–1326, 2006.

[5] G. Chen, K. Mischaikow, R. Laramee, P. Pilarczyk, and E. Zhang, "Vector field editing and periodic orbit extraction using Morse decomposition," *IEEE TVCG*, vol. 13, no. 4, pp. 769–785, 2007.

[6] X. Tricoche, G. Scheuermann, and H. Hagen, "A topology simplification method for 2D vector fields," in *IEEE Vis*, 2000, pp. 359–366.

[7] H. Theisel, T. Weinkauf, H.-C. Hege, and H.-P. Seidel, "Saddle connectors-an approach to visualizing the topological skeleton of complex 3d vector fields," in *IEEE Vis*, 2003, pp. 225–232.

[8] 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 *IEEE Vis*, 2005, pp. 559–566.

[9] Q. Du and X. Wang, "Centroidal voronoi tessellation based algorithms for vector fields visualization and segmentation," in *IEEE Vis*, 2004, pp. 43–50.

[10] A. Telea and J. J. van Wijk, "Simplified representation of vector fields," in *IEEE Visualization*, 1999, pp. 35–42.

[11] Z. Peng, E. Grundy, R. S. Laramee, G. Chen, and N. Croft, "Mesh-driven vector field clustering and visualization: An image-based approach," *IEEE TVCG*, vol. 18, no. 2, pp. 283–298, Feb. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.25

[12] W. de Leeuw and R. van Liere, "Collapsing flow topology using area metrics," in *IEEE Vis.*, 1999, pp. 349–354.

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

[14] J. Helman and L. Hesselink, "Representation and display of vector field topology in fluid flow data sets," *IEEE Computer*, vol. 22, no. 8, pp. 27–36, 1989.

[15] G. Chen, K. Mischaikow, R. Laramee, and E. Zhang, "Efficient Morse decompositions of vector fields," *IEEE TVCG*, vol. 14, no. 4, pp. 848–862, 2008.

[16] A. Szymczak and E. Zhang, "Robust morse decompositions of piecewise constant vector fields," *IEEE TVCG*, vol. 18, no. 6, pp. 938–951, 2012.

[17] G. Chen, Q. Deng, A. Szymczak, R. Laramee, and E. Zhang, "Morse set classification and hierarchical refinement using Conley index," *IEEE TVCG*, vol. 18, no. 5, pp. 767–782, 2012.

[18] L. Sipeki and A. Szymczak, "Simplification of Morse decompositions using morse set mergers," in *Topo-In-Vis 2013*, 2013.

[19] A. Globus, C. Levit, and T. Lasinski, "A tool for visualizing the topology of 3-dimensional vector fields," in *IEEE Vis*, 1991, pp. 33–40.

[20] H. Hauser and E. Gröller, "Thorough insights by enhanced visualization of flow topology," in *Symp. on flow vis*, 2000.

[21] S. Mann and A. Rockwood, "Computing singularities of 3d vector fields with geometric algebra," *IEEE Vis.*, pp. 283–290, 2002.

[22] J. Kasten, I. Hotz, B. R. Noack, and H.-C. Hege, "On the extraction of long-living features in unsteady fluid flows," in *Top. Meth. in Data Analysis and Vis.*, 2011, pp. 115–126.

[23] J. Reininghaus, C. Lowen, and I. Hotz, "Fast combinatorial vector field topology," *IEEE TVCG*, vol. 17, no. 10, pp. 1433–1443, 2011.

[24] T. Klein and T. Ertl, "Scale-space tracking of critical points in 3D vector fields," *Top. Methods in Vis.*, pp. 35–49, 2007.

[25] H. Edelsbrunner, D. Letscher, and A. Zomorodian, "Topological persistence and simplification," *DCG*, vol. 28, pp. 511–533, 2002.

[26] H. Edelsbrunner, D. Morozov, and A. Patel, "The stability of the apparent contour of an orientable 2-manifold," in *Top. Methods in Data Anal.and Vis.* Springer-Verlag, 2010, pp. 27–41.

[27] ——, "Quantifying transversality by measuring the robustness of intersections," *FoCM*, vol. 11, pp. 345–361, 2011.

[28] F. Chazal, A. Patel, and P. Skraba, "Computing well diagrams for vector fields on $R^n$," *App. Math. Letters*, vol. 25, no. 11, pp. 1725–1728, 2012.

[29] B. Wang, P. Rosen, P. Skraba, H. Bhatia, and V. Pascucci, "Visualizing robustness of critical points for 2d time-varying vector fields," *Computer Graphics Forum*, vol. 32, no. 2, pp. 221–230, 2013.

[30] P. Skraba, B. Wang, G. Chen, and P. Rosen, "Robustness-based simplification of 2d steady and unsteady vector fields," *IEEE TVCG*, vol. 21, no. 8, pp. 930 – 944, 2015.

[31] P. Skraba and B. Wang, "Interpreting feature tracking through the lens of robustness," *Top. Methods in Data Anal. and Vis. III*, 2014.

[32] P. Skraba, B. Wang, G. Chen, and P. Rosen, "2d vector field simplification based on robustness," *Proc. Pac. Vis. Symp.*, 2014.

[33] V. Guillemin and A. Pollack, *Differential Topology*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1974.

[34] H. Carr, J. Snoeyink, and U. Axen, "Computing contour trees in all dimensions," *SODA*, pp. 918–926, 2000.

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

[36] H. Bhatia, A. Gyulassy, H. Wang, P.-T. Bremer, and V. Pascucci, "Robust detection of singularities in vector fields," in *Top. Methods in Data Anal. and Vis. III*. Springer Berlin Heidelberg, 2014.

[37] H. Edelsbrunner and E. P. Mücke, "Simulation of simplicity: A technique to cope with degenerate cases in geometric algorithms," *ACM Trans. on Graphics*, vol. 9, no. 1, pp. 66–104, 1990.

[38] "CGAL, Comp. Geom. Algorithms Library," http://www.cgal.org.

[39] P. Shirley and A. Tuchman, *A polygonal approximation to direct scalar volume rendering*. ACM, 1990, vol. 24, no. 5.

[40] M. Floater, "Mean value coordinates," *Computer Aided Geometric Design*, vol. 20, pp. 19–27, 2003.

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

[42] X. Ye, D. Kao, and A. Pang, "Strategy for seeding 3d streamlines," *IEEE Vis.*, pp. 471 – 478, 2005.

[43] X. Tricoche, C. Garth, G. Kindlmann, E. Deines, G. Scheuermann, M. Ruetten, and C. Hansen, "Visualization of intricate flow structures for vortex breakdown analysis," *IEEE Vis.*, pp. 187–194, 2004.

[44] C. S. Yoo, E. S. Richardson, R. Sankaran, and J. H. Chen, "A DNS study on the stabilization mechanism of a turbulent lifted ethylene jet flame in highly-heated coflow," *Proc. Combustion Institute*, vol. 33, no. 1, pp. 1619–1627, 2011.

[45] J. Kasten, T. Weinkauf, C. Petz, I. Hotz, B. R. Noack, and H.-C. Hege, "Extraction of coherent structures from natural and actuated flows," in *Active Flow Control II*. Springer, 2010, pp. 373–387.

[46] M. Jiang, R. Machiraju, and D. Thompson, "Detection and visualization of vortices," in *Vis. Handbook*. Acad. Press, 2005, pp. 295–309.

[47] J. Sahner, T. Weinkauf, and H.-C. Hege, "Galilean invariant extraction and iconic representation of vortex core lines," in *EuroVis*, June 2005, pp. 151–160.

[48] T. Schafhitzel, J. Vollrath, J. P. Gois, D. Weiskopf, A. Castelo, and T. Ertl, "Topology-preserving lambda$_2$-based vortex core line detection for flow visualization," *Comp. Graph. Forum*, vol. 27, no. 3, pp. 1023–1030, 2008.

[49] J. Kasten, J. Reininghaus, I. Hotz, and H.-C. Hege, "Two-dimensional time-dependent vortex regions based on the acceleration magnitude," *IEEE TVCG*, vol. 17, no. 12, pp. 2080–2087, 2011.


**Primoz Skraba** is currently a Senior Researcher at the Jozef Stefan Institute and Assistant Professor of Computer Science at the University of Primorska in Slovenia. He received his Ph.D. in Electrical Engineering from Stanford University in 2009. His main research interests are applications of topology to computer science including data analysis, machine learning, sensor networks, and visualization.

**Paul Rosen** is an Assistant Professor at the University of South Florida with the Department of Computer Science and Engineering. He received his PhD degree from the Computer Science Department of Purdue University. His research interests include data analytics, topological data analysis, human oriented design, and visualization education. He is a member of the IEEE.

**Bei Wang** is a Research Computer Scientist at the Scientific Computing and Imaging Institute, University of Utah. She received her Ph.D. in Computer Science from Duke University in 2010. Her main research interests lie in the theoretical, algorithmic, and application aspects of data analysis and data visualization, with a focus on topological techniques. She is also interested in computational biology and bioinformatics, machine learning and data mining. She is a member of ACM and IEEE.

**Guoning Chen** is an Assistant Professor at the Department of Computer Science at the University of Houston. He earned a PhD degree in Computer Science from Oregon State University in 2009. Before joining the University of Houston, he was a post-doctoral research associate at Scientific Computing and Imaging Institute at the University of Utah. His research interests include visualization, data analytics, computational topology, geometric modeling, and geometry processing, and physically-based simulation. He is a member of ACM and IEEE.

**Harsh Bhatia** is a post-doctoral researcher at the Lawrence Livermore National Laboratory. His research interests include scientific and information visualization, topological analysis, uncertainty visualization, computer graphics, statistical learning, and modeling and simulation. Harsh completed his PhD from the Scientific Computing and Imaging (SCI) Institute, University of Utah in 2015. Prior to grad school, he received the BTech degree in information and communication technology from DA-IICT, India in 2007.

**Valerio Pascucci** is a Professor of the Scientific Computing and Imaging Institute, a Professor of the School of Computing, University of Utah, and a DOE Laboratory Fellow, of the Pacific Northwest National Laboratory. He earned a Ph.D. in computer science at Purdue University in 2000, and a EE Laurea (Master), at the University La Sapienza in Roma, Italy, in 1993. His recent research interest is in developing new methods for massive data analysis and visualization.