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

Primož Skraba, Paul Rosen, Member, IEEE, Bei Wang, Member, IEEE, Guoning Chen, Member, IEEE, 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 is no 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 a 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 non-trivial separation structures, and thus is computationally effective. Furthermore, our algorithm can remove critical points in any subregion 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 method 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, and 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 it 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 little work has been 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 on this topic 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 the approach in [8], we propose the first framework that directly cancels pairs of 3D critical points with a guaranteed minimum amount of perturbation for 3D vector fields. Importantly, our framework 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 Fig. 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, as long as it has zero degree. Our algorithm does not require special boundary configurations.

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, little work has been done on the topology-based simplification in 3D, due at least in part, to the complexity of the topology of general 3D vector fields [3], [19] which consists not only of 1D skeletons but also of 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, which 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. The method of Weinkauf et al. is fundamentally different from our approach. Whereas 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 treated 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 and Ertl [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. Robustness 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.
to 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 pairs 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 $M$ 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 $D \subseteq M \subset \mathbb{R}^m$ 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 $f : M \to \mathbb{R}^m$ to be a vector field on a manifold $M$ (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 $D$ that 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 $u : \partial D \to S^{m-1}$ that maps the boundary of $D$ to the $(m-1)$-sphere, given by $u(x) = f(x)/|f(x)|$ (which 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 $f : M \subset \mathbb{R}^m \to \mathbb{R}^m$, 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 flow, 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 $r_1$, and $x_3$ and $x_4$ have 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 $d(f, g) \leq r$, where $d(f, g) = \sup_{x \in \mathbb{R}^2} ||f(x) - g(x)||_2$. 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 of $f$, such that $h^{-1}(0) \cap D = \emptyset$ and $h = f$ except possibly within the interior of $D$.

**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 $x$, $\deg(h|_{\partial D}) = \deg(f|_{\partial D})$.

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 points in the vector field. A tetrahedron \( t \) (with a critical point \( p \) in its interior) may be assigned a degree of only \(+1\) 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 sign \((ab \cdot (ac \times ad))\), which 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. We use regular tetrahedra \( t \) in the domain, which 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 \( T \) 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 the 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 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 piece. In particular, this scheme would require only the subdivision of triangles that intersect with 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.

Image space. Given a 3D vector field restricted to a degree-zero component \( D \subseteq \mathbb{R}^3 \), \( f : D \to \mathbb{R}^3 \), 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 mapping is similar to the concept of a Gauss map except that the vectors are not normalized. Since \( D \subseteq \mathbb{R}^3 \), it follows that \( \text{im}(D) \) is contained within a 3-ball of radius \( r \), denoted as \( S^2_r \), whose boundary is a 2-sphere, denoted as \( S^2 \). Similar to the 2D case [32], we consider the boundary of \( \text{im}(D) \) uncovered, if \( \partial(\text{im}(D)) \subset S^2 \), 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, that is, \( \partial D \) need not be simply connected or may even consist of multiple components.

4.2 Algorithm Details

Algorithm overview. Given a zero-degree connected component \( D \subseteq \mathbb{R}^3 \), our simplification strategy consists of three steps.
We perform Laplacian smoothing on such that it of a component inside that operation may increase the amount of perturbation, but it is not strictly required for simplification.

Cut operation. Recall that the image space $\text{ima}(D)$ of a component $D \subseteq \mathbb{R}^3$ by construction belongs to a 3-ball $S^3$. The center $O$ of $S^3$ (i.e., the origin) is part of $\text{ima}(D)$ if $D$ contains critical points. The Cut operation then deforms $\text{ima}(D)$ such that there is a small void surrounding the origin, which means the resulting vector field restricted to $D$ do 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 $S^3$. Although a deterministic algorithm exists to locate a valid cut point via stereographic projection, in practice, uniform random sampling on $S^3$ works very 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 $p \in S^3 - \text{ima}(\partial D)$. As illustrated in Fig. 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 $\text{ima}(D)$ 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 $\text{ima}(\partial D)$. 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 $\text{ima}(D)$. 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 $r + \varepsilon$, which represents the bound on the amount of perturbation for simplification. Note that the 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 Figs. 5c, 5d, 5e (please refer to Section 5 for details on the datasets). If the boundary is uncovered as in Fig. 5b, the internal tetrahedra may be projected to uncover the origin and hence remove the critical points.

Unwrap operation for covered image space. The final scenario is that the degree of $D$ is 0, but $\text{ima}(\partial D)$ covers $S^3$. The Cut cannot be directly applied since its corresponding sampling procedure relies on $\text{ima}(\partial D)$ 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 occurs more readily in 3D,
The above procedure is equivalent to Laplacian smoothing restricted to the sphere $S^2_r$. In practice, $\text{im}(\partial D)$ 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 the 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 $\text{im}(\partial D)$ to the sphere $S^2_r$. See Fig. 6 for an illustration.

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 $a$), it can be shown that the re-normalization step does not affect convergence, as it 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 affected 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 supplementary video, available online, for the synthetic dataset) are outlined spheres that move at a constant speed in the forward time direction along the streamlines in order to convey the directions 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 Ye et al. [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 the angle of attack eventually leads to vortex breakdown [43]. The final dataset is the Lifted Flame dataset. It is a sub-volume 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, see our supplemental video, available online.

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 10 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 (indicated by the arrows) have opposite degrees, i.e., degree +1 (red ball) and degree −1 (blue ball). As we increase the threshold of the 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 second and third viewpoints in Fig. 8. Their corresponding visualizations are provided in Fig. 9, where we highlight the location of the critical points before 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 the source, 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 a vector field modification that is local to the sublevel set, which is desirable 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, and 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 the 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 result 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 exists no critical point in the interior of the spiral region, leading to a potentially continuous representation of the vortex core (not shown here). This is desirable for the subsequent identification and visualization of vortices. We notice that the to-be-modified
sublevel set 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 #1. There are two possible ways to cancel this group of critical points. On the one hand, we could cancel the pair #1 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 canceled together. In either case, our method can successfully replace the flow within the sublevel set region with a critical point free flow. 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 example demonstrates again that our robustness-based simplification requires only a 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). The example of Fig. 13 shows that 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, which 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 subvolume of the Lifted Flame dataset for our experiment. The mesh contains 6,000,000 elements.
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 purposes, it is labeled 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 a 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.

5.4 Discussion on Unwrap Scenario for Synthetic#2

The group of critical points in Synthetic#2 is the only complex scenario 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 the robustness value of the Synthetic#1 pair), and 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, because 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) to ensure that no points in the image of the component 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 the image space 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, a region on the surface of the sphere 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 Discussion

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, because our algorithm requires only a 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 the 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 that contains the −1 degree point and one of the +1 degree points. Applying the algorithm on this subset would cancel this pair.

Vortex and vortex core structure. Vortices are important flow structures that are not part of the vector field topology [46], [47], [48], [49]. However, a unified definition of vortices does not exist. Existing vortex core detection techniques are also sensitive to small perturbations or noise in the data, leading to many small and disconnected segments as well as excessively complex vortex core structure, which 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, little work has been done on the simplification of vortex core structure. Our robustness-based simplification can help fill this gap to some extent. In the case of incompressible turbulent flow, we have seen a number of critical points near vortex cores (Figs. 1 and 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 US National Science Foundation (NSF) ACI-1443046. BW and VP were supported by INL 00115847, DE-AC0705ID14517 and DOE NETL. GC was supported by NSF IIS-1352722.

References


Primož Skraba received the PhD degree in electrical engineering from Stanford University in 2009. He is currently a senior researcher at the Jozef Stefan Institute and an assistant professor of computer science at the University of Primorska in Slovenia. His main research interests are applications of topology to computer science including data analysis, machine learning, sensor networks, and visualization.