Enhanced topology-sensitive clustering by Reeb graph shattering

W. Harvey, O. Rübel, V. Pascucci, P.-T. Bremer, and Y. Wang

Abstract Scalar-valued functions are ubiquitous in scientific research. Analysis and visualization of scalar functions defined on low-dimensional and simple domains is a well-understood problem, but complications arise when the domain is high-dimensional or topologically complex. Topological analysis and Morse theory provide tools that are effective in distilling useful information from such difficult scalar functions. A recently proposed topological method for understanding highdimensional scalar functions approximates the Morse-Smale complex of a scalar function using a fast and efficient clustering technique. The resulting clusters (the so-called Morse crystals) are each approximately monotone and are amenable to geometric summarization and dimensionality reduction. However, some Morse crystals may contain loops. This shortcoming can affect the quality of the analysis performed on each crystal, as regions of the domain with potentially disparate geometry are assigned to the same cluster. We propose to use the Reeb graph of each Morse crystal to detect and resolve certain classes of problematic clustering. This provides a simple and efficient enhancement to the previous Morse crystals clustering. We provide preliminary experimental results to demonstrate that our improved topology-sensitive clustering algorithm yields a more accurate and reliable description of the topology of the underlying scalar function.

Oliver Rübel and Peer-Timo Bremer

Valerio Pascucci and Peer-Timo Bremer

William Harvey and Yusu Wang

Ohio State University, 487 Dreese Lab, 2015 Neil Ave., Columbus, Ohio 43210. e-mail: {harveywi, yusu}@cse.ohio-state.edu

Lawrence Livermore National Laboratory, L-422, P.O. Box 808, Livermore, CA 94551-0808. e-mail: {ruebel1, bremer5}@llnl.gov

Scientific Computing and Imaging Institute, School of Computing, University of Utah, 72 South Central Campus Drive, Salt Lake City, Utah 84112. e-mail: {pascucci,ptbremer}@sci.utah.edu

1 Introduction

Scalar-valued functions are abundant in scientific data. Understanding and visualizing the structure of these functions is a fundamentally important aspect of scientific research. When the domain of a scalar-valued function is low-dimensional, these two tasks are often straightforward. However, as the topological and geometric complexity of the domain increases, exploring and understanding the function can become challenging. Recently, feature-based methods using ideas from geometry and topology have proven useful for extracting meaning from high-dimensional data.

Computational efficiency is very important for data exploration, yet it can be an elusive goal when the data is high-dimensional. For example, creating a precise reconstruction of a high-dimensional domain from point cloud data and detecting high-dimensional cycles can be computationally prohibitive. Thus, a reasonable compromise is to strive for a balance between fidelity and processing speed. The fundamental idea of topology-sensitive clustering is to partition the domain into regions that are in some sense simple, capturing the essence of the topological structure of the scalar function at hand while maintaining economy of computing effort.

One may argue that a good clustering algorithm will partition the domain into monotone regions which are approximately topological balls, as this would provide simplicity of both the function behavior and topological structure within each region. Additionally, the accuracy of approximation should be exchanged for a significant gain in computational efficiency.

In this work, we take an additional step toward this ideal topology-sensitive clustering algorithm by extending the method of Gerber et al. [16]. Specifically, [16] describes a simple discrete gradient based method to cluster points into so-called *Morse crystals* approximately analogous to the Morse-Smale complex of the input function. These crystals have simple topology (ideally a topological ball) within each of them, and have been demonstrated to be useful in understanding high dimensional scalar fields in various ways. However, in practice, these Morse crystals may have non-trivial topology inside. In this work, we take advantage of the ability to detect a certain type of non-trivial topology within each crystal, namely one-dimensional cycles (*loops*) reflected in the structure of its Reeb graph. By this property, the Reeb graph is used to subdivide a crystal into pieces which are topologically simpler. We also show preliminary experimental results to demonstrate the effectiveness of the proposed methods in several datasets.

2 Related Work

Topological methods such as the Reeb graph [29] and Morse-Smale complex [25], provide abstract representations of the fundamental structure of scalar functions. Topological structures make complex functions accessible to computational analysis and provide efficient means for defining a wide variety of features. Scalar-field

2

topology as a general tool for analysis of scalar functions has been used in a wide array of applications ranging from, for example, physics [5, 23], biosciences [20], and medicine [7] to material sciences [17].

The Morse-Smale complex is defined as the intersection of the stable and unstable manifolds of f, i.e., the Morse-complex of f and -f. The Morse-Smale complex partitions the domain into regions of monotone gradient flows, each with a single source, defined by a maximum of f, and a sink, defined by a minimum of f. In the area of visualization, computation of Morse-Smale complexes has focused mainly on n-dimensional manifolds [13, 14, 18]. Algorithms for computing the Morse complex — although under different names — have been proposed in a wide range of research areas. In computational geometry, the Morse complex is often described in terms of a filtration of sub-level sets of f [9, 8, 35]. The watershed segmentation method [4, 27], widely used in image processing, is a variant of the Morse-complex and has been described for image data [3] as well as abstract-graphs and n-dimensional grids [34]. Gradient ascent-based clustering methods, such as mean-shift [10, 12], medoid-shift [30], and quick shift [32] clustering, are widely used in machine learning and pattern recognition, and they are also closely related to algorithms for computing and approximating the Morse complex. Our work is based on and extends a recently proposed algorithm by Gerber et al. [16] for approximating the Morse-Smale complex of unstructured, point-cloud data in n-dimensions, described in more detail in Section 3.2.

While the Morse-Smale complex describes the topology of a scalar function f based on the induced gradient flow, the Reeb graph [29] encodes the topology of the set of level-sets of f. The Reeb graph is constructed by contracting the connected components of the level-sets of f to points. In this work we use the Reeb graph to detect and correct loops in crystals of the approximate Morse-Smale complex. In literature, various efficient algorithm for computing the Reeb graph and its loop-free variant, the contour-tree, have been described for n-dimensional manifolds [28, 6, 21]. Here, we use the fast randomized algorithm proposed by Harvey et al. [21] to compute the augmented Reeb graph of a point set.

In order to compute the Morse-Smale complex, information about the connectivity of the domain is needed. Neighborhood/proximity graphs [22] such as the knearest neighbor graph and Delaunay triangulation [2] are commonly used to define neighborhoods for point cloud data. We use the approximate, undirected k-nearest neighbor graph for this purpose. We then construct the 2-skeleton of the domain, required for the Reeb graph computation, by calculating the Rips complex [33] (see Section 3.3.1).

3 Method

Overview. We consider the problem of understanding the structure of a scalar function defined on a high-dimensional domain $f : \mathbf{M} \to \mathbf{R}$, where $\mathbf{M} \subseteq \mathbf{R}^D$. In practice, information about the scalar function is often available only as point cloud data; i.e.,

a set $X = \{x_1, x_2, ..., x_n\}$ of *n* discrete point samples taken from the domain, and their associated set of function values $Y = \{f(x_1), f(x_2), ..., f(x_n)\}$. Thus, the objective is to acquire some insight into the structure of *f* based on properties which can be obtained through the analysis of X and Y. Our algorithm proceeds as follows:

- 1. **Domain Approximation.** The connectivity of the underlying domain, if not available, must be approximated from *X*. We use the approximate, undirected *k*-nearest neighbor graph for this purpose.
- 2. Morse Crystal Decomposition. *X* is partitioned into Morse crystals using the method proposed in [16].
- Reeb Graph Shattering. Crystals containing loops are shattered according to the structure of their Reeb graphs.

3.1 Domain Approximation

Before any topological processing can occur, the structure of the domain must be approximated from X. The proposed method is agnostic with respect to how the approximation is performed; any suitable method which attempts to capture the neighborhood relationships of the point cloud data can be used. For the purposes of applying the technique to analysis of high-dimensional datasets while maintaining computational efficiency, we prefer to use the approximate, undirected *k*-nearest neighbor graph of X to represent the topology of the underlying domain. We compute this graph efficiently using the freely-available ANN library [26].

3.2 Morse Crystal Decomposition

In this section we review the Morse crystal decomposition of f, introduced briefly in [16]. We provide a detailed description of this decomposition and later show in Section 3.3.2 how it is augmented to yield the shattered Morse crystals.

We begin by considering the *k*-nearest neighbors of a point, and we describe how the function values of those neighbors give rise to the Morse crystal decomposition. Without loss of generality, henceforth we assume that all points in *X* have unique function values. Let *x* be a point of *X*, and let $G_{knn} = (X, E_{knn})$ be the undirected knearest neighbor graph of *X*, where $E_{knn} \subset X^2$. Let $n(x) = \{y : (x, y) \in E_{knn}\})$ denote the k-nearest neighbors of *x*. For any neighbor *y* of *x*,

$$m(x,y) = \frac{f(y) - f(x)}{\|y - x\|}$$

describes the approximate gradient of *f* at *x* in the direction (y - x). Let $n^+(x) = \{y \in n(x) : f(x) < f(y)\}$ denote the *upper neighbors of x* and $n^-(x) = \{y \in n(x) : x \in n(x)\}$

f(x) > f(y) denote the *lower neighbors of x*. Then the *steepest ascending neighbor* x^+ and *steepest descending neighbor* x^- of x are defined as

$$x^+ = \operatorname*{arg\,max}_{y \in (n^+(x) \cup \{x\})} m(x, y); \qquad x^- = \operatorname*{arg\,min}_{y \in (n^-(x) \cup \{x\})} m(x, y).$$

The notions of steepest ascending and descending neighbors of a point lead to succinct definitions of the *steepest ascending path of x* as $\pi^+(x) = \{x\} \cup \pi^+(x^+)$ and the *steepest descending path of x* as $\pi^-(x) = \{x\} \cup \pi^-(x^-)$. Finally, let $\pi(x) = \pi^+(x) \cup \pi^-(x)$ denote the points along the discrete integral curve passing through *x*.

Since the Morse crystal decomposition relies on the extremal elements of $\pi(x)$ for each point $x \in X$, we will use them to define an equivalence class that gives rise to the decomposition. Let $\omega^+(x)$ and $\omega^-(x)$ denote the maximal and minimal elements of $\pi(x)$, respectively. Two points *x* and *y* are *equivalent*, denoted $x \sim y$, if their steepest ascending paths converge to a common maximum and their steepest descending paths converge to a common minimum; that is $\omega^+(x) = \omega^+(y)$ and $\omega^-(x) = \omega^-(y)$. Then the quotient space X_{\sim} forms the *Morse crystal decomposition of f*.

Based on these observations, it is apparent that computing X_{\sim} admits an efficient and straightforward solution. Let F^+ and F^- be two disjoint-set forests [31], where F^+ supports $union^+(x,y)$ and $find^+(x)$ operations, and F^- supports $union^-(x,y)$ and $find^-(x)$ operations. For each point *x*, take $union^+(x,x^+)$ and $union^-(x,x^-)$. Then the Morse crystal membership of *x* is uniquely identified as the ordered pair $(find^-(x), find^+(x))$. Figure 1 illustrates the Morse crystal decomposition of point cloud data sampled from a simple two-dimensional domain.

Topological Simplification. To help reduce the number of spurious Morse crystals caused by sparsity in the input data or the presence of noise, the Morse crystal decomposition can be simplified by neutralizing crystals whose topological persistence is below a user-specified threshold τ . Specifically, we can use the union-find



Fig. 1 An example of the Morse crystal decomposition of a simple scalar function. Here, the function value at each point is its height. (a) *k*-nearest neighbor graph of points sampled from a simply connected, lambda-shaped domain. (b) Edges connecting points to their steepest ascending neighbors. (c) Edges connecting points to their steepest descending neighbors. (d) The Morse crystal decomposition partitions the domain into four approximately-monotone regions.

data structure to also compute the standard persistence for each minimum and maximum [15]. The output is a set $S = \{(x_i, k_i, s_i, \delta_i)\} \subset X^3 \times \mathbb{R}^+$ which encodes the persistence and pedigree of the topological components of f. Here, x_i is an extremum of a topological component with persistence δ_i , k_i is the extremum of the component which kills x_i , and s_i is the saddle point merging the two components. By using the map

$$\kappa_{\tau}(x_i) = \begin{cases} k_i \text{ if } \delta_i < \tau \\ x_i & \text{otherwise} \end{cases}$$

topological simplification of the Morse crystal decomposition occurs by simply using $(\kappa_{\tau}(find^{-}(x)), \kappa_{\tau}(find^{+}(x)))$ to assign points to Morse crystals.

3.3 Reeb Graph Shattering

While the Morse crystal decomposition provides a useful segmentation of the domain into approximately piecewise-monotone regions, its computational efficiency trades off fidelity to the true underlying Morse-Smale complex. As a result, a crystal may contain nontrivial loops or higher-order voids. These voids contribute to the geometric disparity of the points within the crystal, which is antithetical to our desired domain segmentation.

We propose to focus on the problem of finding and eliminating certain 1-loops in Morse crystals through the process of *Reeb graph shattering*. That is, we can further refine each Morse crystal by tracking the sequence of Reeb graph edges that these paths visit as they traverse the crystal, and using these sequences to potentially split the crystal into *shattered Morse crystals*. Figure 2 illustrates the process of shattering a loop-containing Morse crystal into a small collection of shattered Morse crystals. We remark that we choose to use the Reeb graph to again trade off fidelity with efficiency — Computing the Reeb graph of a crystal takes near-linear time whereas computing a set of generating 1-cycles of an input crystal takes time cubic in the size of the crystal. (Note that both methods require a 2-skeleton of the crystal as input; in extreme cases this takes time cubic in the size of the crystal to compute.) Furthermore, as we will show below, the use of the Reeb graph also provides a natural and simple way to further segment the input crystal; while in general, subdividing a domain to remove all non-trivial 1-cycles does not appear to be an easy problem.

3.3.1 The Reeb Graph

Let $f: X \to \mathbf{R}$ be a continuous function. A *level set* of f with value $\alpha \in \mathbf{R}$ is the set $f^{-1}(\alpha) = \{x \in X : f(x) = \alpha\}$. Two points $x, y \in X$ are *equivalent with respect* to relation R, denoted xRy, if x and y reside in the same connected component of some level set. The *Reeb graph of* f [29], denoted $\mathscr{R}_f(X)$, is the quotient space X_R .

Intuitively, the Reeb graph captures the merging and splitting behavior of the level sets of f. $\mathscr{R}_f(X)$ has a graph structure, and its *nodes* refers to those points such that either their up-degree or their down-degree is not 1 (where "up" and "down" refer to order in function values). An *arc* of the Reeb graph is a maximal connected component after removing nodes from $\mathscr{R}_f(X)$. See Figure 2(b) for an example, where the Reeb graph of the height function defined on a torus has four nodes and four arcs.

The Reeb graph depends only on the 2-skeleton of its domain. Thus, we must extract this information for each crystal. The Rips complex provides a 2-skeleton that is sufficient for extracting topological information. It is readily computed from G_{knn} by removing edges whose endpoints lie in disparate crystals, and then transforming each 3-clique among the remaining edges into a 2-simplex. The fast randomized algorithm of Harvey et al. [21] is then used to produce the Reeb graph of each crystal.



Fig. 2 (a) Example of a Morse crystal which contains a large loop. For this example, the height function is used. (b) The Reeb graph of this crystal. (c) Regions of the crystal corresponding to the four edges of its Reeb graph. (d) The crystal is partitioned into two shattered Morse crystals according to which regions of (c) each integral curve passes through. Here, the integral curves of the left shattered crystal pass through regions $\{R_1, R_2, R_4\}$, and integral curves of the right shattered crystal pass through regions $\{R_1, R_3, R_4\}$.

3.3.2 Shattered Morse Crystal Decomposition

Let $G_C = (V_C, E_C)$ denote the *k*-nearest neighbor graph of *X* constrained to a Morse crystal $C \subseteq X$ output by [16]. Let $\mathscr{R}(C) = (V_R, E_R)$ be the Reeb graph of *C*. Let *g* map each edge of G_C to its set of corresponding Reeb graph edges. Given a discrete integral curve π , $g(\pi)$ denotes its image (which is also a path) in the Reeb graph $\mathscr{R}(C)$. Two points $x, y \in X$ are *shattered Morse crystal equivalent*, denoted $x \stackrel{\sim}{\sim} y$, if $g[\pi(x)] = g[\pi(y)]$. That is, two points are equivalent if they are in the same original crystal output by [16], and the discrete integral curves passing through them map to the same set of Reeb graph edges. The resulting quotient space X_{\sim} is the *shattered Morse crystal decomposition of f*.

Topological Simplification In practice, $\mathscr{R}(C)$ may contain a variety of lowpersistence topological features (topological noise) that can result in an oversegmentation of the crystal. Fortunately, features arising from spurious bumps and small loops can be eliminated with the help of the *extended persistence* algorithm [1, 11] Specifically, the lowest and highest points of each loop in $\mathscr{R}(C)$ are paired using the algorithm, and if their difference in function value is below a user-specified threshold v, the loop can be removed from $\mathscr{R}(C)$.

Recall that the input crystal *C* itself may be obtained after some topological simplification as described in Section 3.2. Hence it may contain multiple extreme points inside, and an integral path π in *C* may end up at some local min and max, instead of the global min and max. Now after $\mathscr{R}(C)$ is topologically simplified into \mathscr{R}' , it may have fewer extreme than the input crystal *C*. This means that the image of an integral path π in \mathscr{R}' may start and/or end with some interior points, instead of with extreme points in \mathscr{R}' . This causes some technical problem when comparing whether two integral paths are equivalent or not in the simplified Reeb graph. To reconcile this disparity, the notion of steepest ascending (x^+) and descending (x^-) neighbors of a point $x \in C$ is modified to take into account the structure of \mathscr{R}' . Specifically, if *x* is a local maximum in *C* but not in \mathscr{R}' , then its steepest ascending neighbor is defined to be its upper neighbor in the simplified Reeb graph \mathscr{R}' . An analogous redefinition is used for the steepest descending neighbors of local minima.

Computation Given a Morse crystal *C* output from [16], we first compute its Reeb graph $\mathscr{R}(C)$ using the algorithm from [20], and simplify it to \mathscr{R}' . Next, for two points $x, y \in C$, to test for their membership, we need to first compute their corresponding integral paths in *C*, and then check for their images.

To compute their corresponding integral paths $\pi(x)$ and $\pi(y)$, we use the same method as described in Section 3.2 but from the modified forests $T^+ = (C, E^+)$ and $T^- = (C, E^-)$ with $E^+ = \{(x^+, x) : x^+ \neq x\}$ and $E^- = \{(x^-, x) : x^- \neq x\}$. These forests encode the steepest ascending and descending paths of a point in *C* based on the modified definitions of steepest neighbors as introduced above. Next, we need to compute the images $g[\pi(x)]$ and $g[\pi(y)]$. For simplicity, in our current implementation, we compute $g[\pi(x)]$ as the sequence of Reeb graph arcs stabbed by the images of vertices from $\pi(x)$ (instead of as the concatenation of the images of edges from $\pi(x)$)¹. Two points *x* and *y* are equivalent if their integral paths produce the same sequence of Reeb graph arcs in \mathscr{R}' .

4 Results

We begin by investigating a pair of synthetic low-dimensional point cloud datasets. The underlying scalar functions of these datasets are precisely known and they exhibit simple structure, yet their Morse crystal decompositions contain crystals with loops. We show that the shattered Morse crystal decompositions of these datasets

¹ We remark that such sequences produce the same equivalence relations as the one defined earlier when the input points are dense enough so that there is no long edge whose image spans several Reeb graph arcs.

Enhanced topology-sensitive clustering by Reeb graph shattering

successfully resolve the loops. To conclude, we examine a real dataset consisting of points sampled from a scalar function whose domain is a 5-manifold embedded in \mathbf{R}^9 . Our method successfully identifies and corrects problematic crystals in this dataset.



Fig. 3 Simple example of the shattered Morse crystal decomposition of a domain containing a loop. (a) The Morse crystal decomposition of this dataset results in a single crystal containing a large loop. (b) The large crystal is shattered into two parts to eliminate the loop.

4.1 Torus

As a simple example, we illustrate a dataset consisting of 1000 points sampled randomly from a 2-torus of unit diameter, with the scalar value at each point assigned according to the height function (see Figure 3). To approximate the underlying domain, k = 20 is used, and a threshold of $\tau = 0.1$ is used for topological simplification. The (trivial) Morse crystal decomposition is shown in Figure 3(a), i.e., the entire domain is assigned to a single Morse crystal containing a large loop. Figure 3(b) shows the shattered Morse crystal decomposition, which partitions the domain into two regions, neither of which contains the loop of the torus.

4.2 Mixture of Gaussians

The torus example of Section 4.1 exhibits a loop that is attributable to the topology of the domain. However, the algorithm in [16] can produce crystals containing loops even for scalar functions defined on simply connected domains. To demonstrate this phenomenon, we construct a simple example by taking a mixture of five 2D Gaus-



(a)



Fig. 4 Example of loops arising in Morse crystals even when the domain is simply connected. (a) Mixture of five Gaussians. Some discrete integral curves are shown in red. (b) The Morse crystal decomposition results in eight crystals, half of which contain loops. (c) The shattered Morse crystal decomposition successfully resolves the four loop-harboring crystals.

sians (see Figure 4). This function contains a total of eight crystals, half of which contain large loops. The shattered Morse crystal decomposition correctly shatters the crystals with loops, yielding a final segmentation of the domain into twelve regions. Naturally, the Morse crystals which do not contain loops remain intact during the shattering process.



Fig. 5 Optimization dataset visualized by projecting the 9-dimensional points onto their first two principal components. (a) The Morse crystal decomposition consisting of six crystals. (b) A Morse crystal whose Reeb graph contains two prominent loops is partitioned into a small collection of shattered Morse crystals (c).

4.3 Optimization Dataset

In this experiment we demonstrate the applicability of our algorithm to the analysis of the structure of an optimization problem. Given two images with image point correspondences, the goal is to estimate the translation and rotation of two calibrated cameras. This problem can be formulated as a minimization of the *algebraic error* [19] which — given two corresponding points $x = [x_1 x_2 1]^T$ and $x' = [x'_1 x'_2 1]^T$ on the image plane in the respective coordinate frames of the two cameras — can be defined as:

$$x^T E x' = 0. (1)$$

 $E = [t] \times R$ is a 3 × 3 rank-2 matrix, called the *essential matrix*, with the unit vector *t* describing the relative position, or translation, between the two cameras, and the orthogonal rotation matrix *R* representing the relative camera orientation. Both *t* and *R* are expressed in the coordinate frame of *x*. In practice, Eq. 1 can usually not be satisfied exactly due to the presence of noise in *x* and *x'*. The problem is, therefore, formulated as an optimization of the sum of squared residuals

$$f(R,t) = f(E) = \sum_{i} (x_i^T E x_i')^2$$
(2)

over all point pairs $\langle x_i, x'_i \rangle$. In order to minimize f, one needs to determine the 9 elements of E. For more detailed information on the formulation of this problem and how it is solved in practice see the manuscript by Lindstrom and Duchaineau [24].

In order to understand the structure of f we computed 5000 random samples of f in 9-dimensional parameter space. Due to the formulation of the problem we know that E only has 5 degrees of freedom [24], 3 to describe the rotation and 2 to determine the translation up to scale. f, hence, defines a 5-dimensional manifold embedded in 9-dimensional space.

Figure 5 shows the Morse crystal decomposition of the 5000 point samples using 17 nearest neighbors. For these images, the 9-dimensional points have been projected onto their first two principal components. Figure 5(b) reveals a Morse crystal in this dataset whose Reeb graph contains two prominent loops (a topological double-torus). By applying the proposed Reeb graph shattering technique, the crystal is split into subregions.

4.4 Time Complexity

Since the shattered Morse crystal decomposition requires the 2-skeletons of the Morse crystals, it can become expensive to compute if k is chosen to be large (requiring, in the worst case, time cubic in the number of vertices). However, choosing a large k is not necessary in practice since G_{knn} is only meant to capture the topology *locally* around each point. In our experience, combining a reasonably small k (e.g. 10 < k < 40) with topological simplification techniques works well for capturing the important topological features of the dataset at hand while minimizing the prohibitive time complexity of 2-skeleton construction.

5 Conclusions

Reliable estimation of topological structures of high-dimensional functions is essential for accurate topology-based visualization and analysis. Feature-based visualizations and statistical analyses of high-dimensional functions based on the analysis of Morse crystals commonly rely on the notion that a Morse crystal is monotone and has genus 0. Morse crystals with a common source (maximum) and sink (minimum), however, are not separated correctly by the approximate Morse-Smale complex algorithm, which can lead to the creation of Morse crystals with a genus ≥ 1 .

We described a simple and novel algorithm to augment the previous Morse crystal decomposition algorithm to detect and correct certain falsely merged Morse crystals containing possibly multiple loops. Using the per-crystal Reeb graph we detect crystals containing loops and shatter them into multiple, loop-free crystals.

We illustrated the problem of crystals containing loops using two analytic examples and showed that our algorithm produces a correct complex in both cases. We used our algorithm to analyze the structure of a real-world optimization problem, demonstrating its practical relevance. Our algorithm produces a more accurate approximation of the Morse-Smale complex in high-dimensions.

While the proposed algorithm is an improvement, it is still lacking in some areas. Specifically, it is only capable of detecting and resolving those loops which are evident from the structure of the Reeb graph. Thus, it may fail to detect some loops in the domain which do not induce loops in the Reeb graph. Additionally, like its predecessor, it is incapable of dealing with any cycles of dimension 2 or greater that may be present in the dataset. Additional research in topology-sensitive clustering methods will strive to provide insight into how to grapple with these difficulties.

Computation of topological structures of point-cloud data rely on proper estimation of the structure of the domain. The study and development of methods for constructing optimal neighborhood/proximity graphs, hence, promises to enable more accurate approximations of topological structures. In future we also plan to investigate further applications of the approximate Morse-Smale complex to improve understanding of the structure of high-dimensional scalar functions.

Acknowledgements We would like to thank Peter G. Lindstrom for providing us with the optimization dataset and his help and insight into the problem. This work was funded by the National Science Foundation (NSF) under grants CCF-0747082, DBI-0750891, and CCF-1048983. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. We would like to thank the Livermore Elks for their scholarship support. Publication number: LLNL-CONF-468782.

References

- Agarwal, P.K., Edelsbrunner, H., Harer, J., Wang, Y.: Extreme elevation on a 2-manifold. In: Proceedings of the Twentieth Annual Symposium on Computational Geometry, SOCG '04, pp. 357–365 (2004)
- Aurenhammer, F.: Voronoi diagramsa survey of a fundamental geometric data structure. ACM Computing Surveys 23(3) (1991)
- Beucher, S.: Watersheds of functions and picture segmentation. In: Proc. IEEE Int. Conference on ICASSP, pp. 1928–1931 (1982)
- Beucher, S., Lantujoul, C.: Use of watersheds in contour detection. In: Int. workshop on image processing, real-time edge and motion detection (1979)
- Bremer, P.T., Weber, G., Pascucci, V., Day, M., Bell, J.: Analyzing and tracking burning structures in lean premixed hydrogen flames. IEEE Trans. Visualization and Computer Graphics 16(2), 248–260 (2010)
- Carr, H., Snoeyink, J., Axen, U.: Computing contour trees in all dimensions. Comput. Geom. Theory Appl. 24(3), 75–94 (2003)
- Carr, H., Snoeyink, J., van de Panne, M.: Simplifying flexible isosurfaces using local geometric measures. In: IEEE Visualization '04, pp. 497–504. IEEE Computer Society (2004)
- Chazal, F., Guibas, L., Oudot, S., Skraba, P.: Analysis of scalar fields over point cloud data. In: SODA '09: Proc. 19th Annual ACM-SIAM Symp. on Discrete Algorithms, pp. 1021–1030 (2009)
- Chazal, F., Guibas, L., Oudot, S., Skraba, P.: Persistence-based clustering in riemannian manifolds. Tech. Rep. RR-6968, INRIA (2009)
- Cheng, Y.: Mean shift, mode seeking, and clustering. IEEE Trans. Pattern Analysis and Machine Intelligence 17(8), 790–799 (1995)

- Cohen-Steiner, D., Edelsbrunner, H., Harer, J.: Extending persistence using poincaré and lefschetz duality. Foundations of Computational Mathematics 9(1), 133–134 (2009)
- Comaniciu, D., Meer, P.: Mean shift: A robust approach toward feature space analysis. IEEE TPAMI 24, 603–619 (2002)
- Edelsbrunner, H., Harer, J., Natarajan, V., Pascucci, V.: Morse-Smale complexes for piecewise linear 3-manifolds. In: Proc. 19th Symp. on Computational Geometry, pp. 361–370 (2003)
- Edelsbrunner, H., Harer, J., Zomorodian, A.: Hierarchical Morse-Smale complexes for piecewise linear 2-manifolds. Discrete Comput. Geom. 30, 87–107 (2003)
- Edelsbrunner, H., Letscher, D., Zomorodian, A.: Topological persistence and simplification. Discrete Comput. Geom. 28, 511–533 (2002)
- Gerber, S., Bremer, P.T., Pascucci, V., Whitaker, R.: Visual exploration of high dimensional scalar functions. IEEE Trans. Visualization and Computer Graphics 16(6), 1271–1280 (2010)
- Gyulassy, A., Duchaineau, M., Natarajan, V., Pascucci, V., E.Bringa, Higginbotham, A., Hamann, B.: Topologically clean distance fields. IEEE Trans. Visualization and Computer Graphics 13(6), 1432–1439 (2007)
- Gyulassy, A., Natarajan, V., Pascucci, V., Hamann, B.: Efficient computation of Morse-Smale complexes for three-dimensional scalar functions. IEEE Trans. Visualization and Computer Graphics 13(6), 1440–1447 (2007)
- Hartley, R., Zisserman, A.: Multiple View Geometry, second edn. Cambridge University Press (2003)
- Harvey, W., Wang, Y.: Generating and exploring a collection of topological landscapes for visualization of scalar-valued functions. Computer Graphics Forum 29(3), 9931002 (2010)
- Harvey, W., Wang, Y., Wenger, R.: A randomized O(m log m) time algorithm for computing reeb graphs of arbitrary simplicial complexes. In: Proc. Annual Symp. on Computational Geometry 2010, SoCG '10, pp. 267–276. ACM, New York, NY, USA (2010)
- Jaromczyk, J.W., Abstract, G.T.T.: Relative neighborhood graphs and their relatives. Proc. IEEE 80(9), 1502–1517 (1992)
- Laney, D., Bremer, P.T., Mascarenhas, A., Miller, P., Pascucci, V.: Understanding the structure of the turbulent mixing layer in hydrodynamic instabilities. IEEE Trans. Visualization and Computer Graphics 12(5), 1052–1060 (2006)
- Lindstrom, P., Duchaineau, M.: Factoring algebraic error for relative pose estimation. Tech. Rep. LLNL-TR-411194, Lawrence Livermore National Laboratory (2009)
- Morse, M.: Relations between the critical points of a real functions of n independent variables. Trans. of the American Mathematical Society 27, 345–396 (1925)
- 26. Mount, D., Arya, S.: ANN: Approximate Nearest Neighbors. URL http://www.cs.umd. edu/~mount/ANN
- Najman, L., Schmitta, M.: Watershed of a continuous function. Signal Processing, Mathematical Morphology and its Applications to Signal Processing 38(1), 99–112 (1994)
- Pascucci, V., Scorzelli, G., Bremer, P.T., Mascarenhas, A.: Robust on-line computation of reeb graphs: simplicity and speed. ACM Trans. Graph. 26(3), 58 (2007)
- Reeb, G.: Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique [on the singular points of a complete integral pfaff form or of a numerical function]. Comptes Rendus Acad.Science Paris 222, 847–849 (1946)
- Sheikh, Y., Kahn, E., Kanade, T.: Mode-seeking by medoidshifts. In: Proc. IEEE Int. Conference on Computer Vision (2006)
- Tarjan, R.E., van Leeuwen, J.: Worst-case analysis of set union algorithms. J. ACM 31, 245– 281 (1984)
- 32. Vedaldi, A., Soatto, S.: Quick shift and kernel methods for mode seeking. In: Proc. European Conference on Computer Vision (2008)
- Vietoris, L.: Über den höheren Zusammenhang kompakter Räume und eine Klasse von zusammenhangstreuen Abbildungen. Mathematische Annalen 97(1), 454472 (1927)
- Vincent, L., Soille, P.: Watersheds in digital spaces: An efficient algorithm based on immersion simulations. IEEE Trans. Pattern Analysis and Machine Intelligence 13(6), 583–598 (1991)
- Zhu, X., Sarkar, R., Gao, J.: Shape segmentation and applications in sensor networks. In: Proc. INFOCOM, pp. 1838–1846 (2007)

14