

25th International Meshing Roundtable

Computing shape descriptors based on vector-valued functions.

Federico Iuricich^{a,*}, Sara Scaramuccia^b, Claudia Landi^c, Leila De Florian^b

^a*Department of Geography and UMIACS, University of Maryland, College Park (MD), USA.*

^b*Department of Computer Science, Bioengineering, Robotics, and Systems Engineering, University of Genova, Genova, Italy.*

^c*Department of Sciences and Methods for Engineering DISMI, University of Modena and Reggio Emilia, Modena-Reggio Emilia, Italy*

Abstract

We present a new algorithm for computing a discrete gradient field on multivariate data. For multivariate data, we consider a shape with a vector-valued function f defined on it. The proposed algorithm is well suited for parallel and distribute implementations. The discrete gradient field V we obtain is a reduced representation of the original shape Σ (i.e. composed by fewer elements than Σ) and can be used for capturing the relationships among the different scalar functions of f . Moreover, V is proven to have the same multidimensional persistence of Σ .

© 2016 The Authors. Published by Elsevier Ltd.

Peer-review under responsibility of the organizing committee of IMR 25.

Keywords: Shape Analysis, discrete Morse theory, multivariate data

1. Introduction

Topological persistence has proved to be a promising tool for shape analysis. The persistent diagram is a powerful descriptor capturing meaningful features of a shape under study. Generally speaking, shape descriptors are extracted from a shape using multiple functions capturing different properties of the shape. The problem of comparing two meshes is then reduced to comparing the corresponding shape descriptors. Instead of considering each function independently one could group them in a single vector-valued function. Building a descriptor for a multivariate function allows to represent features that are generally missed when working with the functions one by one. While the concepts of persistent homology and persistence diagram for multivariate data are still a matter of study, our contribution focuses on retrieving a compact representation for the shape having the same multidimensional persistence, namely the discrete gradient field. Discrete Morse theory [1], a combinatorial counterpart of Morse theory, provides the notion of discrete gradient field (also called Forman gradient) for an efficient and derivative-free analysis of a scalar field. In the case of univariate data, the Forman gradient has been extensively used because, due to its discrete nature, it can be easily represented [2]. We consider Discrete Morse Theory for computing a discrete gradient field on a shape on which a vector-valued function has been defined. The focus of our work is in practical applications. We provide the

* Corresponding author. Tel.: +0-000-000-0000 ; fax: +0-000-000-0000.

E-mail address: iurif@umd.edu

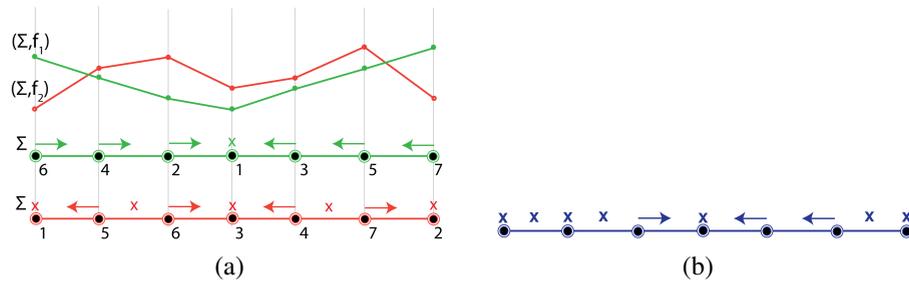


Fig. 1. (a) Two filtering functions f_1 and f_2 defined on the same simplicial complex Σ . The graph of both functions is shown and the values of both f_1 and f_2 are explicitly indicated for each vertex of Σ . The discrete gradient computed on each function is also indicated. Arrows represent gradient pairs and exes represent critical simplices. (b) The discrete gradient computed on (Σ, f) according to the filtering function $f = (f_1, f_2)$. Arrows represent gradient pairs and exes represent critical simplices.

first algorithm capable of computing a discrete gradient field on real-world data. Our approach is easy-to-use and well suited for parallel and distributed implementations.

2. Background notions

In this section, we describe our setting. We will discuss our work in terms of triangle meshes, although, all the results are valid for any kind of regular cell complex.

A k -dimensional simplex, or k -simplex, σ is the convex hull of $k + 1$ affinely independent points. A *face* τ of σ is the convex hull of any subset of $k - 1$ points of σ (indicated $\tau < \sigma$), while σ is a coface of τ (indicated $\sigma > \tau$). A simplicial complex Σ is a collection of k -simplices such that every face of a simplex in Σ is also in Σ and the intersection of any two simplices of Σ is a face of both. Triangle meshes (or meshes for brevity) are examples of simplicial complexes: the vertices, edges, and faces correspond to 0-, 1- and 2-simplices, respectively. We will denote by Σ_k the set of k -simplices in Σ .

Given a triangle mesh Σ , a filtration is a sequence of meshes $\Sigma^0 \subset \Sigma^1 \subset \dots \subset \Sigma^n = \Sigma$. By defining a function on the vertices of Σ it is possible to induce a filtration by assigning, to each simplex of Σ , the maximum function value of its vertices. The filtration of Σ is then defined as the sequence of sub-level complexes $\Sigma^u = f^{-1}(-\infty, u]$.

In Figure 1a we show two different functions defined on the same mesh Σ . Filtering Σ according to f_1 means sweeping the graph of f_1 , from bottom to top, introducing simplices along the way. By starting from the lowest function value, we introduce vertex 1 first. Then, vertex 2 is introduced with edge $\{2, 1\}$ and so on.

In this context, persistent homology [3] is used to study the homological changes of the sub-level sets $\Sigma^u = f^{-1}(-\infty, u]$. Looking at the filtration induced by function f_1 , the only homological change occurs when vertex 1 is introduced, creating the first component. The filtration induced by f_2 instead provokes much more homological changes. Simplices responsible for said changes are indicated with an ex. At the time vertices 1, 2 and 3 are introduced they all originate a new component. Successively vertices 4, 5 and 6 are all introduced with edges $\{3, 4\}$, $\{1, 5\}$ and $\{6, 3\}$ without affecting the homology. By adding edge $\{5, 6\}$ two distinct components become connected, changing the homology again. The same holds for edge $\{4, 7\}$.

In our work, we consider multiple filtering functions all defined on the same dataset. A multifiltration can be seen as an n -dimensional array where the indices of each cell of the array correspond to the coordinates at which a simplex is introduced. For example, considering the bi-filtration $f = (f_1, f_2)$, the leftmost vertex of Figure 1a enters at coordinates $(6, 1)$. The adjacent vertex enters at coordinates $(4, 5)$ and the edge in between appears at coordinates $(6, 5)$ (i.e., when both its vertices are introduced). Assigning each simplex to a cell of the matrix, based on their coordinates, the reader can build a matrix of all the simplicial complexes arising from the bi-filtration. In this context, multidimensional persistent homology aims at detecting the homological changes among nested pairs of complexes. While referring to [4] for a precise definition of multidimensional persistence, for the sake of this work we consider it as the generalization of persistence homology for a multifiltration.

The algorithm proposed in this paper retrieves an acyclic discrete vector field (called discrete gradient for brevity) over the domain Σ . The relevance of this output has to be seen within the framework of Forman's discrete Morse

Theory [1]. In discrete Morse Theory, a (*discrete*) *vector* is a pair of simplices (σ, τ) such that $\sigma < \tau$. A *discrete vector field* V is any collection of vectors over a simplicial complex such that each simplex belongs to at most one vector. Given a discrete vector field V , if a simplex belongs to no vector, it is called *critical*. A V -path is a sequence of vectors (σ_i, τ_i) belonging to V , for $i = 1, \dots, r$, such that, for all indexes $i \leq r - 1$, $\sigma_{i+1} < \tau_i$ and $\sigma_i \neq \sigma_{i+1}$. A V -path might be *closed* if $\sigma_1 = \sigma_r$ and *trivial* if $r = 1$. A *discrete gradient* V is a discrete vector field whose closed V -paths are all trivial. Forman [1] proves that the homology of Σ is always isomorphic to the homology of V . The discrete gradient V can be adapted to preserve the sub-level structure with respect to a filtering function. In the univariate setting, Theorem 4.3 in [5] proves that the persistent homologies of V also coincides with those of Σ . In Figure 1a we depict with an arrow the gradient pairs of the Forman gradient V computed for each filtering function. In this case, we can notice that the unpaired (critical) simplices are exactly those responsible for a topological change. This means that computing persistent homology, using all the vertices and edges of Σ or using only the simplices marked by an ex, lead to the same result. For multidimensional persistent homology, this result is guaranteed by Corollary 3.12 in [6]. Referring to Figure 1a, an example of discrete gradient compatible to $f = f(f_1, f_2)$ is shown in Figure 1b.

3. Computing the discrete gradient

The first attempts for extending the concept of discrete gradient to the multivariate case can be found in [6,7]. Both put the theoretical foundation inspiring also our work but, they are not computationally feasible especially when working on real-world data.

We consider a triangle mesh Σ and a vector-valued function $f : \Sigma_0 \rightarrow \mathbb{R}^n$ defined on its vertices Σ_0 . The function f is required to be *component-wise injective*. Moreover, we extend the function f by defining for each simplex $\sigma \in \Sigma$ and each component i from 1 to n , $\tilde{f}_i : \Sigma \rightarrow \mathbb{R}^n$, $\tilde{f}_i(\sigma) := \max_{v \in \sigma} f_i(v)$. The proposed algorithm takes direct inspiration from the one presented in [8] for scalar fields. Similarly to the latter, our output will consists of two lists V and C containing the list of pairs (σ, τ) , and the list of the critical simplices, respectively. As in [8], simplices belonging to the same sublevel sets are paired via homotopy expansion. Two simplices, say k -simplex σ and $(k + 1)$ -simplex τ , are paired via homotopy expansion when σ has no unpaired faces and τ has only one unpaired face (i.e. σ).

As a first step, a suitable indexing is computed for defining a total order over the vertices of Σ . We require the indexing I to be *well-extendible* with respect to the function \tilde{f} , i.e., for every two vertices v_1 and v_2 , if $\tilde{f}(v_1) \leq \tilde{f}(v_2)$, then $I(v_1) \leq I(v_2)$. I is then extended by defining for each simplex $\sigma \in \Sigma$, $\tilde{I}(\sigma) := \max_{v \in \sigma} I(v)$. The algorithm processes the vertices one by one (possibly in parallel) and, for each vertex $v \in \Sigma_0$, the index-based lower star $L_I(v)$ is computed.

$$L_I(v) := \{\sigma \in \Sigma \mid \tilde{I}(\sigma) = I(v)\}. \quad (1)$$

Each index-based lower star $L_I(v)$ can possibly consists of a single simplex, which has to be the vertex v itself, or different simplices. The union of all the index-based lower stars is a partition of Σ .

The simplices in $L_I(v)$ are then subdivided according to \tilde{f} . More precisely, simplices in $L_I(v)$, having the same function value \tilde{f} , are grouped together in set S . For each set S we compute the local discrete gradient (V_S, C_S) via homotopy expansion. Each local pair in V_S and each critical simplex in C_S contributes to the global output (V, C) in an independent way. At this point, computing the gradient pairs via homotopy expansion has no conceptual differences from the one described in [8] except that we will work with the simplices in S only.

In Figure 2, we can see a working example for the procedure `ComputeDiscreteGradient`. Figure 2a shows the simplicial complex indicating the function \tilde{f} for each simplex and the computed indexing I . In Figure 2b the index-based lower stars are extracted. Simplices having the same value belong to the same lower star $L_I(v)$. Notice that a single $L_I(v)$ may enclose simplices with different values of \tilde{f} . For example, the lower star $Low_I([3])$ contains simplices with values $(3, 0)$, $(3, 1)$, $(3, 2)$ and $(3, 3)$. `HomotopyExpansion` is called independently for each equivalence class in $L_I([3])$ (Figure 2c). The set of critical simplices and gradient pairs obtained from each lower star are combined in the final discrete gradient depicted in Figure 2d.

The resulting discrete gradient is proved to have the same *multidimensional persistence* [4] of the original triangle mesh complex. The formal proof is omitted for brevity.

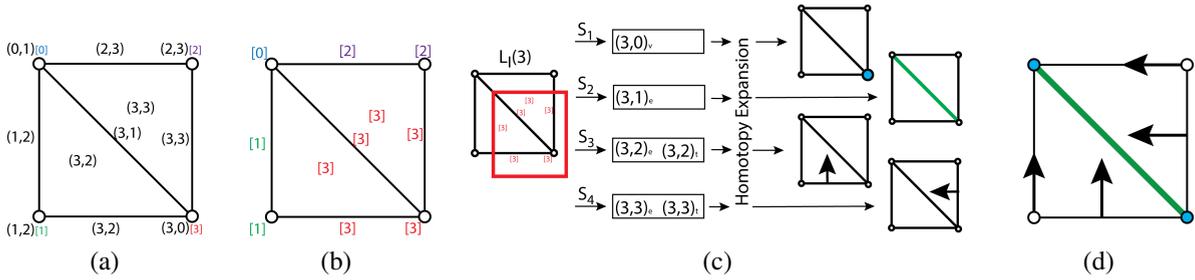


Fig. 2. (a) The filtering function values are depicted as pairs. The vertex indexing I is depicted within square brackets. (b) \tilde{I} is propagated to all the simplices. Colors indicate simplices belonging to the same index-based lower star. (c) For each lower star, the level sets S_i are built collecting simplices having the same function value \tilde{f} . Pairs and critical simplices are created via homotopy expansions within each S_i . (d) The discrete gradient is obtained as the union of all the pairs and critical simplices classified in the lower stars. Critical vertices are depicted in blue and the critical edge is depicted in green.

4. Results and Applications

The proposed algorithm allows for a fast and efficient computation of a discrete gradient when working with multivariate data. This is promising especially for computing multidimensional persistent homology. By computing the multidimensional persistent homology one means retrieving the persistence module [4] The algorithm proposed in [9] for computing the persistence module of a multifiltration has worst time complexity $O(n^4 m^3)$, where n is the number of functions and m is the number of simplices. Other targets for the multidimensional persistent homology computation are the *rank invariant* and the *multigraded Betti numbers* [4] having complexity $O(m^{2n})$. The tool RIVET [10] is an interesting optimized visualization tool for the $n = 2$ case. The tool constructs a suitable *barcode template* in $O(m^3 \kappa + (m + \log \kappa) \kappa^2)$, where $\kappa = \kappa_1 \kappa_2$ with κ_i the number of different coordinates for the i^{th} -component in the support of the multigraded Betti numbers of any index. Then, the tool manages to complete the rest of the information by updating those results in linear time with respect to m .

By removing unnecessary cells, our contribute is twofold. On the one hand, it allows reducing the impact of parameter m in a single and consistent way in multidimensional persistent homology computation without the need of repeating the procedure for each slice. Moreover, for the rank invariant, entire function level sets might disappear and this possibly reduces the impact of parameter κ . Experiments have been performed on a MacBook Pro with a 2.8GHz quad-core processor and 16GB of memory. The preliminary results have been obtained on three triangle meshes of big size (from 12 to 84 millions of simplices), each having three scalar fields defined on the vertices (see [11] Section 6.2 - Db2 for details on the functions). For each dataset, we have studied the performances in computing the discrete gradient V . We have also evaluated the timings for navigating the gradient paths of V , a fundamental operation for computing multidimensional persistence. Computing the discrete gradient takes 2.1 minutes on the smaller dataset and 5.6 minutes on the bigger one. Navigating the gradient paths requires more time, taking 7.3 minutes on the smaller mesh and 22.9 on the bigger one. The efficiency provided by our algorithm in computing the discrete gradient is remarkable. We have also implemented a multithreaded version of the same algorithm using OpenMP, achieving a 3x speedup. For each dataset we have studied the compression factor, i.e. the ratio between the number of simplices in the original mesh and the number of critical simplices in the discrete gradient. We have registered a compression factor between 15x and 8x in the worst case. Clearly, this result strongly depends on the functions chosen for describing the dataset. The more they are similar the less critical simplices are identified and the higher is the compression factor.

In Figure 3 two examples of multivariate data are shown. In Figure 3a three functions are computed on the same mesh. Critical simplices forming the obtained discrete gradient are indicated as colored dots (vertices-blue, edges-green, triangles-red). The same framework is used considering a single function computed on three different poses of the same mesh. The result is shown in Figure 3b. The clusters of critical simplices are an intuitive estimator for identifying areas where the multiple functions disagree. As can be seen in Figure 3 the number of critical simplices is much lower than the total number of simplices thus candidating the method to be a promising preprocessing step for multidimensional persistence.

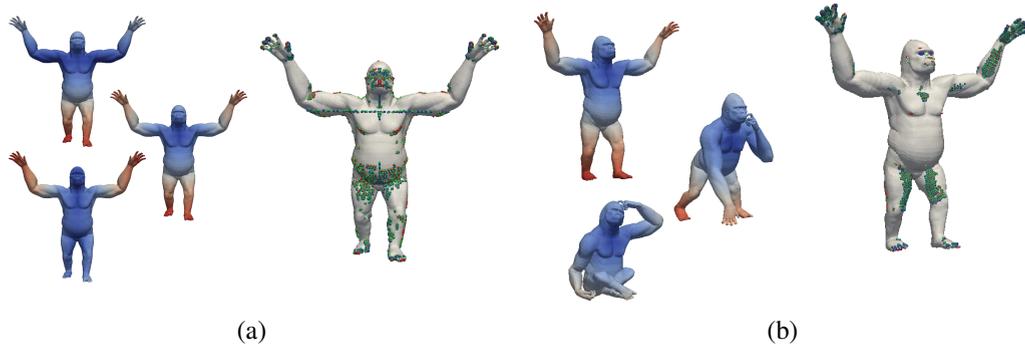


Fig. 3. Two examples of multivariate datasets for shape analysis. (a) The vector-valued function is obtained from three different scalar functions computed on the vertices of the same triangle mesh. (b) The vector-valued function is obtained from a single scalar function computed on the vertices of a mesh in three different poses.

5. Research activity and future works

The new algorithm proposed for computing a discrete gradient vector field on multivariate data is promising to be a useful tool for shape analysis of the future. The algorithm is easy to parallelize and the actual performances have been tested on real-world datasets proving its practical importance. By proving the equivalence with the output of [7] we have demonstrated that the discrete gradient computed is compatible with the multidimensional persistent homology induced by the multiple functions. Thus, our contribution represents the first step towards an efficient computation of the multidimensional persistent homology. Taking advantage from the dimension-agnostic framework of discrete Morse theory we are currently using said algorithm for studying a different kind of data such as weather simulations (2D images) and scientific simulations (3D images). Taking inspiration from what has been done in the univariate case we are also studying the information retrieved by visiting the gradient V-paths. Since the huge number of critical cells typically identified in the multidimensional case it will be fundamental to study a simplification process to reduce their number.

References

- [1] R. Forman, Morse theory for cell complexes, *Advances in mathematics* 134 (1998) 90–145.
- [2] L. De Floriani, U. Fugacci, F. Iuricich, P. Magillo, Morse Complexes for Shape Segmentation and Homological Analysis: Discrete Models and Algorithms, *Computer Graphics Forum* 34 (2015) 761–785.
- [3] H. Edelsbrunner, J. Harer, Persistent homology—a survey, *Contemporary mathematics* 453 (2008) 257–282.
- [4] G. Carlsson, A. Zomorodian, The theory of multidimensional persistence, in: *SoCG '07 Proceedings of the twenty-third annual symposium on Computational geometry*, volume 392, ACM New York, Gyeongju, South-Korea, 2007, pp. 184–193. doi:10.1145/1247069.1247105.
- [5] K. Mischaikow, V. Nanda, Morse Theory for Filtrations and Efficient Computation of Persistent Homology, *Discrete & Computational Geometry* 50 (2013) 330–353.
- [6] M. Allili, T. Kaczynski, C. Landi, Reducing complexes in multidimensional persistent homology theory, *Journal of Symbolic Computation* (2016).
- [7] M. Allili, T. Kaczynski, C. Landi, F. Mazoni, A new matching algorithm for multidimensional persistence, Preprint ArXiv (2015).
- [8] V. Robins, P. J. Wood, A. P. Sheppard, Theory and algorithms for constructing discrete morse complexes from grayscale digital images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33 (2011) 1646–1658.
- [9] G. Carlsson, G. Singh, A. Zomorodian, Computing multidimensional persistence, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 5878 LNCS (2009) 730–739.
- [10] M. Lesnick, M. Wright, Interactive Visualization of 2-D Persistence Modules, Preprint ArXiv (2015) 1–75.
- [11] A. Cerri, B. D. Fabio, G. Jablonski, F. Medri, Comparing shapes through multi-scale approximations of the matching distance, *Computer Vision and Image Understanding* 121 (2014) 43–56.