Approximating lower-star persistence via 2D combinatorial map simplification

Guillaume Damiand [a], *, Eduardo Paluzo-Hidalgo [b], Ryan Slechta [c], Rocio Gonzalez-Diaz [b]

[a] CNRS, LIRIS, UMR5205, Université de Lyon, Lyon 69622, France
[b] Universidad de Sevilla, Dpto. de Matemática Aplicada I, S-41012, Spain
[c] Department of Computer Science and Engineering, The Ohio State University, Columbus, OH 43210, USA

Keywords:
Persistent homology computation
2D combinatorial map
Mesh simplification

A B S T R A C T

Filtration simplification consists of simplifying a given filtration while simultaneously controlling the perturbation in the associated persistence diagrams. In this paper, we propose a filtration simplification algorithm for orientable 2-dimensional (2D) manifolds with or without boundary (*meshes*) represented by 2D combinatorial maps. Given a lower-star filtration of the mesh, faces are added into contiguous clusters according to a “height” function and a parameter ϵ. Faces in the same cluster are merged into a single face, resulting in a lower resolution mesh and a simpler filtration. We prove that the parameter ϵ bounds the perturbation in the original persistence diagrams, and we provide experiments demonstrating the computational advantages of the simplification process.

1. Introduction

Topological data analysis (TDA) is a relatively new subfield of computer science. One of the most useful tools in TDA is *persistent homology*, which is an algebraic method for associating topological features with discrete data. In particular, persistent homology requires two crucial components: (1) a *cell complex* denoted K to structure the data; and (2) a *filtration*, or a nested sequence of indexed subcomplexes where the initial complex is the empty complex and the terminating complex is K. See [1–3] for initial reports and [4] for a modern exposition of the field.

Much work has been done to simplify complexes and filtrations in order to quickly compute persistence. In [5], the authors proposed an efficient algorithm that computes persistent homology for 3D gray-scale images by first obtaining the Morse-Smale complex, which is much smaller than the input complex K but contains all necessary information. The authors first computed a combinatorial gradient vector field using an algorithm from [6], which induces a Morse-Smale complex. The persistence given by this new complex is then exactly the same as that given by the input complex. More recently, in [7], the authors first simplify a filtration of an arbitrary simplicial complex with “strong collapses”. The persistence associated with the reduced filtration is then the same as the original one.

Unlike the previous approaches, we approximate the persistence given by a filtration associated with some 2D mesh (i.e. a piecewise-linear orientable 2D manifold with or without boundary) within a user specified tolerance ϵ. This is accomplished by merging faces in the mesh that meet a criterion. Other authors have taken similar approaches. In [8], the authors gave two different approaches for approximating Čech persistence by building an approximation to the Čech filtration. In [9], the authors developed a notion of (p, ϵ)-admissible, where upon contracting a (p, ϵ)-admissible edge in some complex K, the difference in p-dimensional persistence diagrams given by the initial and the simplified filtration is bounded above by ϵ. We attain a similar goal: upon merging clusters of faces which meet our criterion, called ϵ-permissible, the difference in the 0-, 1-, and 2-dimensional persistence diagrams given by the lower-star filtration on the initial and “merged” meshes is bounded by ϵ.

In [10], the authors proposed an efficient algorithm for computing the homology of meshes represented by 2D combinatorial maps, thereby avoiding the time-consuming step of constructing and modifying boundaries and coboundaries of cells. The process consists of merging faces if they share a common edge, guaranteeing that the structure of the combinatorial map and the homology of the mesh is preserved throughout the process.

This paper extends the work in [10,11] by giving an algorithm to approximate the lower-star persistent homology of meshes within a specified tolerance ϵ. First, faces are grouped into clusters according to the parameter ϵ. Faces in the same cluster are subsequently merged. In [11], cluster membership was determined

* Corresponding author.
E-mail addresses: guillaume.damiand@liris.cnrs.fr (G. Damiand), epaluzo@us.es (E. Paluzo-Hidalgo), slechta.3@osu.edu (R. Slechta), rogodi@us.es (R. Gonzalez-Diaz).

by a distance between faces, and there was no theoretical guarantee that persistence diagrams corresponding to the initial and simplified filtrations were "close." In this paper, membership is determined by the faces' "height", and as our main contribution, we provide such a guarantee in Section 4. Experiments in Section 5 demonstrate the utility of our approach, and we conclude with a brief discussion of possible directions for future work in Section 6.

2. Preliminary notions

In this section we review combinatorial maps and persistent homology.

2.1. 2D combinatorial maps

Roughly speaking, a 2D combinatorial map [12,13], called a 2-map, is a representation of a mesh (see Fig. 1). A mesh M is a 3-tuple (V, E, F) where V is a set of vertices or 0-cells, E is a set of edges or 1-cells, and F is a set of faces or 2-cells.

Cells are in relation together. Two cells are said to be *incident* if one cell belongs to the boundary of the other. Two i-cells are *adjacent* if there exists a $(i-1)$-cell incident to both. Two i-cells are *neighbors* if they are both in the boundary of an $(i+1)$-cell.

Some cells with specific properties are identified. An edge is *dangling* if it is incident to a vertex which is incident to no other edge. An edge is *isolated* if it has no adjacent edge. An edge is *inner* if it is incident to two different faces. An edge is a *border* if it is contained in the boundary of the mesh.

A *2D combinatorial map* (or simply, a 2-map) is a 3-tuple (D, β_1, β_2) where D is a finite set of darts and β_1 and β_2 are one-to-one mappings D onto D, and $\beta_2 = \beta_2^{-1}$.

Meshes can be represented by 2-maps as follows. A dart d represents an orientation of an edge; $\beta_1(d)$ is the dart following d and belonging to the same face as d; $\beta_2(d)$ denotes the opposite orientation of d. In Fig. 1(b), darts 8 and 11 represent the two possible orientations of a single edge, so $\beta_2(8) = 11$ and $\beta_2(11) = 8$.

A dart belongs to exactly one vertex, one edge and one face, and thus each cell of the mesh is described by a set of darts in the 2-map. For example, in Fig. 1(b), vertex v_1 is described by the set of darts {2, 5, 8, 12}. Even isolated edges (like e_7 in Fig. 1(a)) belong to a degenerate face (which is why there are 5 faces in Fig. 1(a)). Inner, dangling, and isolated edges are described always by two darts d_1, d_2. They can be detected in a 2-map thanks to particular configurations of darts and β links (for example an edge $\{d_1, d_2\}$ is isolated if $\beta_1(\beta_1(d_1)) = d_1$).

A border edge is described by one dart d (like dart 16 in Fig. 1(b)). In such a case, $\beta_2(d) = \emptyset$ and β_2 is a partial bijection.

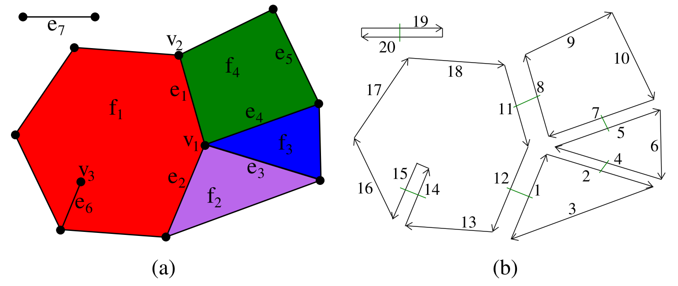


Fig. 1. (a) Example of a mesh with 5 faces (the four faces incident to vertex v_1, and a "degenerate" face bounded twice by edge e_7), 14 edges (e_6 is dangling, e_7 is isolated, $\{e_1, e_2, e_3, e_4\}$ are inner edges and the rest are border edges) and 12 vertices. (b) The corresponding 2-map has 20 darts. Images taken from [10].

2.2. Persistent homology

In this subsection we review persistent homology, an important tool used to assign structure to discrete data. We assume that the reader has some knowledge of homology (for a reference, see [14]). For additional information on persistent homology, we encourage the reader to consult [4].

Persistent homology captures the topological changes occurring in a growing sequence of meshes $M_1 \subseteq M_2 \subseteq \ldots \subseteq M_n = M$, called a *filtration*. As one progresses through the filtration, homology classes of different dimensions may appear (be born) and disappear (die). Some homology classes may be present in M_α as $\alpha \to \infty$. Such classes give the homology of M.

Filtrations are frequently constructed on a mesh M via a real-valued function h on the vertices of M. One such example is the *lower-star filtration* $\{M_\alpha\}_{\alpha\in\mathbb{R}}$ where a cell σ is in M_α if for all vertices v which are incident to σ, $h(v) \le \alpha$. Intuitively, this filtration corresponds to taking sublevel sets of M under h.

If a homology class is born at M_i and dies entering M_j then $j-i$ is the *persistence* of the homology class. If it is born at M_i but never dies then its persistence is set to infinity. Intuitively, homology classes with low persistence are topological noise and the ones that persist correspond to features of the mesh.

The information obtained when computing persistent homology of a filtration can be summarized by a set of *persistence diagrams*, each of which is a multi-set of (birth, death) pairs in the extended real plane, where each pair represents the birth and death of a homology class. For a filtration $\mathcal{M} := \{M_\alpha\}_{\alpha\in A}$, there is a persistence diagram for each dimension p denoted $\mathrm{Dgm}_p(\mathcal{M})$. As expected, $\mathrm{Dgm}_p(\mathcal{M})$ contains persistence information for p-dimensional homology classes. In addition, all points on the diagonal given by $y = x$ are considered to be in each persistence diagram with infinite multiplicity. This enables comparing diagrams with a different number of homology classes with positive persistence. One of the most popular tools for comparing persistence diagrams is the *bottleneck distance.*

Definition 1. The bottleneck distance between two persistence diagrams $\mathrm{Dgm}_p(\mathcal{A})$ and $\mathrm{Dgm}_p(\mathcal{B})$ is

$$d_b(\mathrm{Dgm}_p(\mathcal{A}), \mathrm{Dgm}_p(\mathcal{B})) := \inf_{\psi\in\Psi} \sup_{a\in\mathrm{Dgm}_p(\mathcal{A})} ||a - \psi(a)||_\infty$$

where Ψ is the set of bijections from $\mathrm{Dgm}_p(\mathcal{A})$ to $\mathrm{Dgm}(\mathcal{B})$.

From now on, by abuse of notation, we use M to denote a mesh, the 2-map representing the mesh, or the filtration considered on the mesh. In this paper, persistence diagrams are computed via Algorithm 1, which is a modification of the incremental algorithm given in [15]. Given a mesh and an ordering of its cells, Algorithm 1 computes a 3-tuple (M, H, f) where M is the

Algorithm 1 Computing persistent homology (Algorithm 2 of [15]).

```
Input: An ordering of the cells of $M$: $\{\sigma_1, \ldots, \sigma_m\}$.
Output: Persistent homology of $M$ with respect to such ordering.
Initialize $H \leftarrow \emptyset$ and $f(\sigma_i) \leftarrow 0$, for $1 \le i \le m$.
for $i = 1$ to $m$ do
    if $f\partial(\sigma_i) = 0$ then
        $f(\sigma_i) \leftarrow \sigma_i$  $H \leftarrow H \cup \{\sigma_i\}$(a new homology class was born)
    if $f\partial(\sigma_i) \neq 0$ then
        Let $\sigma_j \in f\partial(\sigma_i)$, $j = \max\{\mathrm{index}(\mu) : \mu \in f\partial(\sigma_i)\}$ $H \leftarrow H \setminus \{\sigma_j\}$(an old homology classdied) foreach $x \in M$ such that $\sigma_j \in f(x)$ do
            $f(x) \leftarrow f(x) + f\partial(\sigma_i)$.
```

given mesh represented by a 2-map; H is the set of *surviving cells*; and f is a map from the k-cells in M to a sum of surviving cells. For a k-cell σ, $\partial(\sigma)$ denotes the set of $(k-1)$-cells in its boundary. Algorithm 1 guarantees that the set of all the surviving k-cells (for a fixed k), together with the disjoint union of sets operation $+$, forms the group $C_k(H)$ which is isomorphic to $H_k(M)$. In addition, the map $f: C_k(M) \to C_k(H)$ satisfies the property that if a, $b \in C_k(M)$ are two homologous k-cycles then $f(a) = f(b)$.

Let M_{σ_i} denote the set of cells $\{\sigma_1, \ldots, \sigma_i\}$ considered in the ith step of Algorithm 1. Note that σ_i belongs to a k-cycle c in $C_k(M_{\sigma_i})$ if and only if $f \circ \partial(\sigma_i) = 0$. Consequently, if $f \circ \partial(\sigma_i) = 0$ then a new homology class was born (represented by c) and σ_i is added to H. Otherwise, if $f\circ\partial(\sigma_i) \neq 0$, then a homology class died, and an element of $f\circ\partial(\sigma)\subseteq H$ is removed from H. The element removed from H is the "youngest" one: argmax{ index(μ): $\mu \in f\circ\partial(\sigma_i)$ }, where index(μ) denotes the position of the cell μ in the ordered list of cells $\{\sigma_1, \ldots, \sigma_m\}$.

The authors in [16] subsequently established a correspondence between the incremental algorithm for computing AT-models given in [15] and the one for computing persistent homology in [4]. The complexity of Algorithm 1 is $O(m^3)$, m being the number of cells in M.

3. Approximating persistence

We now move to describing our procedure for approximating the persistence diagram of the lower-star filtration of a mesh M. The procedure follows three main steps: (1) simplification of the 2-map according to a parameter ϵ; (2) computation of the lower-star filtration of the simplified mesh; (3) computation of persistent homology of the given filtration.

In step 1, we simplify the input 2-map by merging faces. When two faces are merged, the number of cells in the 2-map is reduced, and hence the lower-star filtration corresponding to the reduced 2-map contains fewer cells. By merging several faces, we hope to drastically reduce the number of cells in the reduced filtration to accelerate the runtime for computing persistence in step 3. Note that since the initial and reduced filtrations are different, the persistence is almost certainly different. To ensure that the bottleneck distance between persistence diagrams corresponding to the initial and reduced filtrations is within a user specified tolerance ϵ, we will require merged faces to satisfy a particular condition.

We pay particular attention to step 1. In this step, the 2-map is simplified by dispatching the faces into clusters subject to some constraints according to the parameter ϵ. Note that since h is only defined on the vertex set of M, the index at which a face enters the lower-star filtration is precisely the maximum height value of a constituent vertex. Hence, for face σ, we use the notation $\mathrm{maxh}(\sigma) := \max\{h(v) \mid v \text{ is a vertex of } \sigma\}$ to denote the height, while $\mathrm{minh}(\sigma)$ is defined as expected. We will ensure that all clusters of size greater than 1 are ϵ-permissible.

Definition 2. A cluster of faces $C = \{\sigma_1, \ldots, \sigma_m\}$ is ϵ-permissible if $\mathrm{maxh}(C) - \mathrm{minh}(C) \leq \epsilon$ where $\mathrm{maxh}(C) = \max_{\sigma\in C}\{\mathrm{maxh}(\sigma)\}$ (analogously for minh).

We partition M into ϵ-permissible clusters via Algorithm 2. The first face of each cluster is chosen arbitrarily, and the rest are added using neighborhood relations satisfying ϵ-permissibility. If σ is not yet assigned into a cluster, we create a new cluster C_k and assign σ to C_k. Then, while C_k is ϵ-permissible, C_k is grown by progressively adding faces which are adjacent to the cluster. Each cluster is associated with a value corresponding to the minimum minh of all faces in the cluster and a value corresponding to the maximum maxh of all faces in the cluster, which makes it easy to check if adding a new face would violate the ϵ-permissible condition.

Algorithm 2 Finding ϵ-permissible clusters

Input: A 2-map M; a parameter ϵ
Output: A set of ϵ-permissible clusters $\Lambda = \{C_1, \ldots, C_n\}$.

```
$k \leftarrow 1$
foreach face $\sigma$ of $M$ not yet assigned in a cluster do
    add a new cluster $C_k$ to $\Lambda$;
    add $\sigma$ into $C_k$;
    $\mathrm{T} \leftarrow$ all faces adjacent to $\sigma$;
    while $\mathrm{T}$ is non empty do
        $\tau \leftarrow$ one face in $\mathrm{T}$;
        remove $\tau$ from $\mathrm{T}$;
        if $\tau$ is not yet assigned in a cluster and
        $\max\{\mathrm{maxh}(\tau), \mathrm{maxh}(C_k)\} - \min\{\mathrm{minh}(\tau), \mathrm{minh}(C_k)\} \leq \epsilon$ then
            add $\tau$ in $C_k$;
            add all faces adjacent to $\tau$ into $\mathrm{T}$;
    $k \leftarrow k + 1$;
```

Algorithm 3 Simplification of a cluster.

Input: A 2-map K representing the cluster.
Output: The simplified 2-map corresponding to K.

```
foreach edge $e$ of $K$ do
    if $e$ is an inner edge then remove $e$;
    else
        while $e$ is dangling do
            $e' \leftarrow$ one edge adjacent to $e$
            remove $e$;  $e \leftarrow e'$.
```

We repeat this process until each face is assigned into a cluster. After dispatching all of the faces to clusters, we apply Algorithm 3 which simplifies the 2-map by merging each cluster. The simplification process is depicted in Fig. 2. Critically, the result of merging faces via Algorithm 3 always results in a topological disk with a connected boundary. The complexity of Algorithm 3 is $O(n\alpha(n))$, where n being the number of darts in the 2-map and $\alpha(n)$ is the slow growing inverse Ackermann function (cf. [10,17] for more details). Note that since $\alpha(n) \leq 5$ for all practical purposes, Algorithm 3 is functionally linear.

Once the mesh is simplified, the only remaining steps are to compute the lower-star filtration of the simplified mesh, and then to compute the persistent homology using Algorithm 1.

The complexity of the whole process is $O(n + O(m^3))$, n being the number of darts of the given 2-map, and m being the number of cells in the simplified 2-map. This shows the interest of simplifying the 2-map before computing the persistent homology.

4. Persistence diagram stability

Let M be a polygonal mesh represented as a 2-map, and let M' denote the mesh obtained by merging clusters as explained in Section 3. In this section, we move to showing that merging ϵ-permissible clusters bounds the perturbation in the persistence diagrams associated with the lower-star filtration. Throughout this section, we will use $\mathrm{Dgm}_p(M)$ to denote the p-dimensional persistence diagram given by the lower-star filtration of M. We will also use the notation M_α to denote $\{\sigma \in M \mid \mathrm{maxh}(\sigma) \leq \alpha\}$. In the interest of simplicity, we abuse notation and do not distinguish between vertices in M and M'. If v is a vertex in M', then there is a natural corresponding vertex in M. We will use v to refer to both of these vertices. Likewise for edges and faces. In addition,

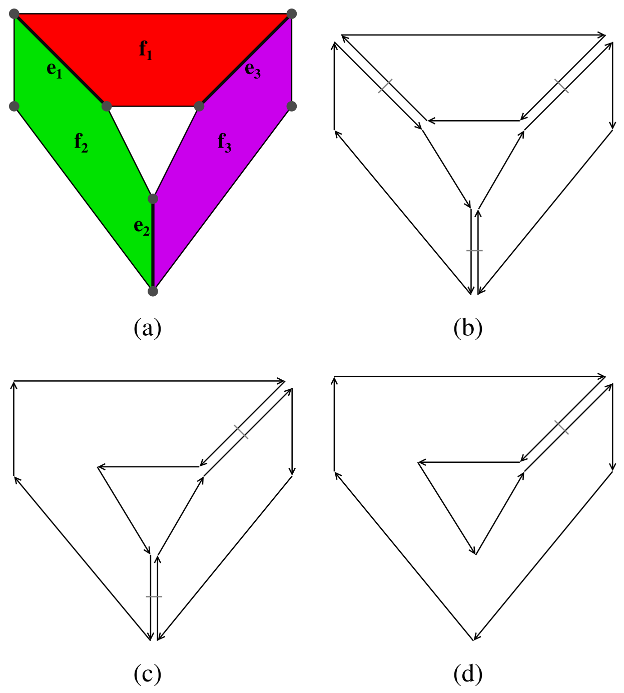


Fig. 2. (a) A cluster composed of three faces with three highlighted inner edges (e_1, e_2, and e_3). (b) The corresponding combinatorial map. (c) The inner edges e_1 has been removed. (d) The inner edge e_2 has been removed. In this last case, e_3 is no longer an inner edge, and hence is not removed by cluster merging.

we will assume that there is a total order $\prec$ on the simplices in M which respects the height function h and dimension. That is, for $\sigma, \tau \in M$, if $h(\sigma) < h(\tau)$, then $\sigma \prec \tau$. Similarly, if $h(\sigma) = h(\tau)$ and $\dim(\sigma) < \dim(\tau)$, then $\sigma \prec \tau$. Hence, for each set of simplices in M, there exists a unique minimal simplex. This total order allows us to consider all those simplices in M which precede simplex σ under $\prec$. We denote such a set as $M_{\prec\sigma}$.

Lemma 1. *Let $u, v \in M$ be vertices which are also in M', where M' is obtained by merging a collection of ϵ-permissible clusters. If u and v are in the same connected component in M_a, then u and v are in the same connected component in $M'_{a+\epsilon}$.*

Proof. Note that clusters are merged into a single face which is homeomorphic to a disk. Hence, if u and v are connected in M_a by a path of edges which includes edges that are removed by cluster merging, then the segment of edges between two vertices on the boundary of a cluster can be replaced by instead traversing the boundary of the cluster. If w is a vertex removed by cluster merging and which is also the face of an edge that connected u and v, then all edges e on the boundary of the cluster containing w must have height $h(e) \le h(w) + \epsilon \le a + \epsilon$. Hence, there is a path between u and v in $M'_{a+\epsilon}$. $\square$

Theorem 1. *If M' is obtained from M by merging a collection of ϵ-permissible clusters, then $d_b(\mathrm{Dgm}_0(M), \mathrm{Dgm}_0(M')) \le \epsilon$.*

Proof. We proceed by constructing a matching $\gamma : \mathrm{Dgm}_0(M) \to \mathrm{Dgm}_0(M')$. Consider $(b, d) \in \mathrm{Dgm}_0(M)$. The point (b, d) corresponds to the lifetime of some homology class created by vertex $v \in M$. We use the notation $[v]_M$ to refer to the set of vertices in M which are in the same connected component as v prior to the introduction of the edge which kills the homology class created by v at index d (if there is such an edge). If $d = \infty$, then $[v]_M$ refers to those vertices in the same connected component as v in M. For $v' \in M'$, we define $[v']_{M'}$ in an analogous way.

In order to define γ, we consider two cases. First, we assume that d is finite. In this case, the homology class $[v]_M$ dies when an edge e is introduced to M_d such that the vertices v and u are in the same connected component, where $u \prec v$. We associate with v, u a unique vertex in M', denoted v', u'. In particular, we let v' denote the minimal vertex in $[v]_M \cap M'$ under $\prec$. If there is no such v', then $[v]_M$ only includes vertices which are removed via face merging. Hence, $[v]_M$ dies when an edge $e = \{w_0, w_1\}$ joins the connected component containing v with the one containing u. In particular, both w_0 and w_1 must be in the cluster containing v, as if w_1 is not in the cluster, then w_0 must be on the boundary of the cluster containing v which would imply that v' exists. Ergo, by ϵ-permissibility, $h(v) = b \le h(e) = d \le h(v) + \epsilon$. Hence, we let $\gamma(b, d) = \left(\frac{b+d}{2}, \frac{b+d}{2}\right)$, which satisfies $\|\gamma(b, d) - (b, d)\|_\infty = \max\left\{\left|\frac{b-d}{2} - b\right|, \left|d - \frac{b-d}{2}\right|\right\} = \max\left\{\left|\frac{-b-d}{2}\right|, \left|\frac{d-b}{2}\right|\right\} = \left(\frac{d-b}{2}\right) \le \epsilon/2$.

We let u' be the minimal element in the set $[u]_M \cap M_{\prec e} \cap M'$. If u' does not exist, then when $[u]_M$ merges with $[v]_M$, the connected component $[u]_M$ contains only vertices which are removed during cluster merging. By identical reasoning as in the case where v' was undefined, the connected component $[v]_M$ must contain a vertex w which is in the same cluster as u. Hence, as $[u]_M$ is joined to $[v]_M$ upon the introduction of an edge e in M_d, where e is in the cluster containing u, it follows that $h(u) \le h(e) \le h(u) + \epsilon$ by ϵ-permissibility. Therefore, as $h(v) \le h(e)$, it follows that $h(e) - h(v) = d - b \le \epsilon$. Hence, we let $\gamma(b, d) = \left(\frac{b+d}{2}, \frac{b+d}{2}\right)$, which we have already shown to satisfy $\|\gamma(b, d) - (b, d)\|_\infty \le \epsilon/2$.

Hence, we assume that both u' and v' exist. By Lemma 1, the vertices u', v' are in the same connected component in $M'_{d'}$, $d' \le d + \epsilon$. It follows that $d \le d'$, else u and v would be in the same connected component prior to M_d. Note that u', v' satisfy $h(u) \le h(u') \le h(u) + \epsilon$ and $h(v) \le h(v') \le h(v) + \epsilon$. This is because if u is not removed, then $u' = u$ and $h(u) = h(u')$. If u is removed and u' exists, then the connected component containing u prior to joining with v must contain some vertex w on the boundary of the merged cluster, which satisfies $h(u) \le h(w) \le h(u) + \epsilon$, and since $h(u) \le h(u') \le h(w)$, it follows that $h(u) \le h(u') \le h(u) + \epsilon$. The same reasoning applies to v, v'.

Now, we consider two subcases. First, we assume that the class $[v']_{M'}$ merges into some other class $[w]_{M'}$ in subcomplex M'_D where $d \le D \le d' \le d + \epsilon$. In such a case, we let $\gamma(b, d) = (h(v'), D)$. Since $b \le h(v') \le b + \epsilon$, and $d \le D \le d + \epsilon$, it follows that $\|\gamma(b, d) - (b, d)\|_\infty \le \epsilon$. Second, we assume that there exists no such D. However, in $M'_{d'}$, v' and u' are in the same connected component, so it follows that $[u']_{M'}$ merges into $[v']_{M'}$. Since by assumption $h(u) \le h(v)$ and $h(v') \le h(u')$, and we have shown that $h(u) \le h(u') \le h(u) + \epsilon$, it follows that $h(u) \le h(v) \le h(v') \le h(u') \le h(u) + \epsilon$, which implies that $0 \le h(u') - h(v) \le \epsilon$. Hence, we let $\gamma(b, d) = (h(u'), d')$ which satisfies $\|\gamma(b, d) - (b, d)\|_\infty \le \epsilon$. These subcases are illustrated in Fig. 3.

In the second case, we assume $d = \infty$. In M', it follows that v' creates a class $[v']_{M'}$ with lifetime $(h(v'), \infty)$. We let $\gamma(b, d) = (h(v'), \infty)$. We have established that either $h(v') = h(v)$ or $0 \le h(v') - h(v) \le \epsilon$. If we have the former, then $\|\gamma(b, d) - (b, d)\|_\infty = 0$. In the event of the later, it follows that $\|\gamma(b, d) - (b, d)\|_\infty \le \epsilon$.

We now prove that γ is a matching. When d is finite, if $\gamma(b_1, d_1) = \gamma(b_2, d_2) = (b', d')$, then (b', d') has multiplicity at least 2. Otherwise, a single class was killed twice, which is a contradiction. In the case where d is infinite, each connected component in M corresponds to a unique connected component in M', so γ must be one-to-one.

To see that γ is onto, first note that every point with infinite persistence is the image of a point with infinite persistence, as cluster merging does not destroy connected components. Second, all $(b', d') \in M'$, $d' < \infty$ correspond to the lifetime of some class created by vertex v' denoted $[v']_{M'}$ which is killed when it merges

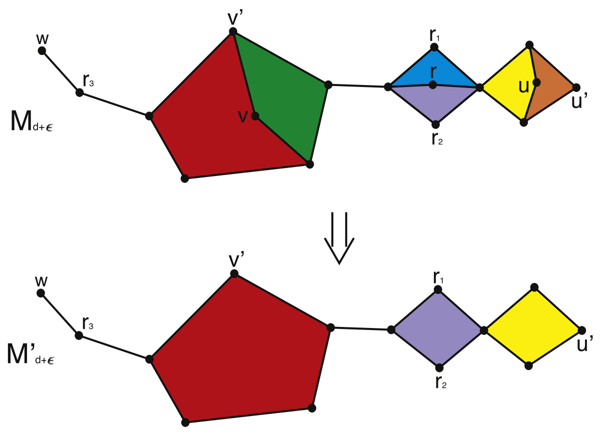


Fig. 3. Let $(b,\ d)$ correspond to the lifetime of $[v]_M$, which in M_d is combined in the same class as u, where $h(u) \le h(v)$. We assume that this happens upon the introduction of vertex r, where $d = h(r) < h(r_1), h(r_2), h(r_3)$, and by ϵ-permissibility, $h(r_1), h(r_2) \le h(r) + \epsilon$. In addition, we let $h(w) < h(v)$. In M', v' and u' (resp. v' and w) are in the same connected component following the introduction of the vertex r_1 (resp. r_3). Hence, if $h(r_1) < h(r_3)$, and $h(u') < h(v')$, then $\gamma(b,d) = (h(v'), h(r_1))$. Similarly, if $h(r_1) < h(r_3)$ but $h(v') < h(u')$, then $\gamma(b,d) = (h(u'), h(r_1))$. If $h(r_3) < h(r_1)$ and $h(w) < h(v')$, then $[v']_{M'}$ is absorbed by $[w]_{M'}$ and not $[u']_{M'}$. Then, $\gamma(b,d) = (h(v'), h(r_3))$. In all cases, $||\gamma(b,d) - (b,d)||_\infty \le \epsilon$.

with some class created by u' denoted as $[u']_{M'}$. Likewise, $[u']_M$ and $[v']_M$ must merge in M at subcomplex M_d. Hence, if $[u']_M$ is the class that dies when merging with $[v']_M$, and (b, d) corresponds to the lifetime of $[u']_M$, then $\gamma(b,d) = (b', d')$. Similarly for if $[v']_M$ dies. Hence, γ is onto.

Thus, γ is a matching where $||\gamma(b,d) - (b,d)||_\infty \le \epsilon$ for all $(b,d) \in \mathrm{Dgm}_0(M)$. Ergo, $d_b(\mathrm{Dgm}_0(M), \mathrm{Dgm}_0(M')) \le \epsilon$. □

Although we could have attempted to give a proof using classical results from [18,19] related to stability of persistence diagrams, we preferred to provide a constructive proof giving the explicit matching between the points in the respective diagrams.

Theorem 2. *If M' is obtained by merging a collection of ϵ-permissible clusters, then $d_b(\mathrm{Dgm}_1(M), \mathrm{Dgm}_1(M')) \le \epsilon$.*

Proof. Analogously to the proof of Theorem 1: if an edge e creates a 1-cycle $\pi = \sum_i e_i$, then we use $[\pi]_M$ to refer to the cycle class containing π. As in the 0-dimensional case, π' refers to the representative cycle of $[\pi]_M$ that is created at the lowest height value and for which none of the edges are removed when clusters are merged. The matching γ is constructed in the same way as the 0-dimensional case: the points corresponding to those classes for which a representative cycle is contained entirely in a cluster and for which the persistence is less than ϵ are paired with the diagonal, points corresponding to those cycles with lifetimes with infinite persistence are paired with canonical choice, and points with finite persistence which do not satisfy the first case are either paired with the diagonal or with the point corresponding to $[\pi']_{M'}$, or, in the event that $[\pi']_{M'}$ kills $[\rho']_{M'}$ where $[\rho]_M$ kills $[\pi]_M$, then the point corresponding to $[\pi]_M$ is paired with the point corresponding to $[\rho']_{M'}$.

The only substantive difference is that in the 0-dimensional case, if $[u]_M$ and $[v]_M$ merge in M_d, then $[u]_{M'}$ and $[v]_{M'}$ merge at $M_{d'}$ where d' satisfies $d \le d' \le d + \epsilon$. In the 1-dimensional case, if $[\pi]_M$ and $[\rho]_M$ merge in M_d, then $[\pi']_{M'}$ and $[\rho']_{M'}$ merge in $M'_{d'}$ where $d - \epsilon \le d' \le d$, as merged faces take the maximum value of the remaining vertices. But this still permits $||\gamma(b,d) - (b,d)||_\infty \le \epsilon$. The function γ is then a matching by the same reasoning. □

Theorem 3. *If M' is obtained by merging a collection of ϵ-permissible clusters, then $d_b(\mathrm{Dgm}_2(M), \mathrm{Dgm}_2(M')) \le \epsilon$.*

Table 1
Average of the bottleneck distance between the persistent diagrams computed on: (1) the lower-star filtration, and (2) the lower-star filtration for different values of Φ for each of the six meshes in (a), and for the six images in (b).

(a)					
Φ	0.32	0.64	1.28	2.56	5.12
ϵ	1.59	3.17	6.34	12.68	25.36
0-D	1.24	2.17	3.89	6.77	9.87
1-D	1.46	2.82	5.37	9.54	23.06
(b)					
Φ	0.32	0.64	1.28	2.56	5.12
ϵ	1.97	3.93	7.87	15.74	31.47
0-D	0.00	2.57	7.28	14.56	27.41
1-D	0.00	2.57	7.28	14.14	26.13

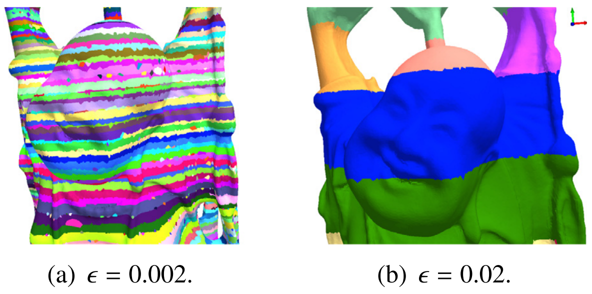
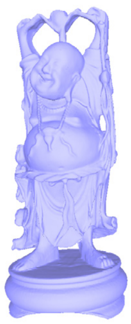
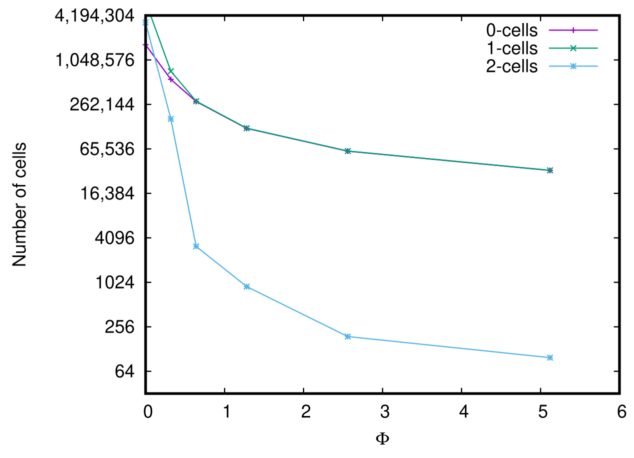


Fig. 5. Effect of the ϵ parameter on the size of the different clusters for the HappyBuddha mesh, zoomed in on the head.

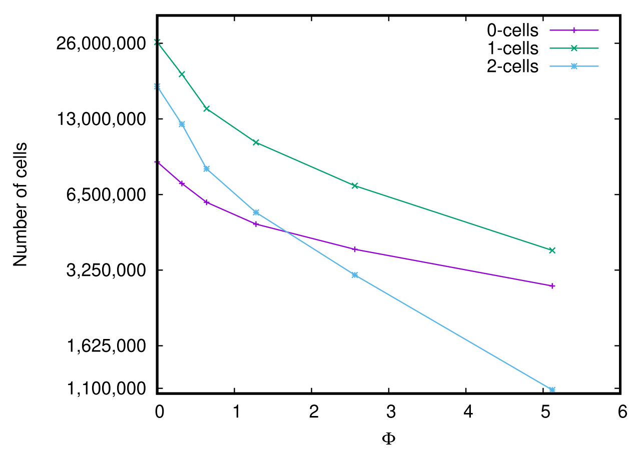


	#0-cells	#1-cells	#2-cells	#*H0*	#*H1*	#*H2*
Blade	882,954	2,648,082	1,765,388	295	330	295
DrumDancer	1,335,436	4,006,302	2,670,868	1	0	1
Neptune	2,003,932	6,011,808	4,007,872	1	6	1
HappyBuddha	543,652	1,631,574	1,087,716	1	208	1
Iphigenia	351,750	1,055,268	703,512	1	8	1
ThaiStatue	4,999,996	15,000,000	10,000,000	1	6	1

Fig. 4. Example of one mesh used in our experiments: the HappyBudda mesh. The table gives the number of i-cells, #i-cells, for the six meshes used, and the number of Hi generators, #Hi, for $i = 0, 1, 2$.



(a)



(b)

Fig. 6. Number of vertices, edges and faces of the simplified 2-maps (in $\log_2$ scale) depending on the value of Φ. These numbers are average values for the six meshes in (a), and for the six images in (b).

Proof. This proof follows analogously to the proof of Theorem 1, except the only case that needs to be considered in when $d = \infty$. Since cluster merging does not destroy 2-cycles or 2-classes, then for each $(b, d) \in \mathrm{Dgm}_2(M)$ there is a canonical choice of a point $(b', d') \in \mathrm{Dgm}_2(M')$, where $b - \epsilon \leq b' \leq b$ and $d = d' = \infty$. Hence, $||\gamma(b, d) - (b, d)||_\infty \leq \epsilon$. □

5. Experiments

We have implemented our algorithm for lower-star filtration simplification by using the CGAL implementation of combinatorial maps (see [20]) and an additional layer, called linear cell complex, which represents the geometry. All experiments were run on an Intel®i7-4790 CPU, 4 cores @ 3.60 GHz with 32 GB RAM. All the computation times given here are averages of 10 consecutive runs on the same mesh.

5.1. 3D meshes

In our first experiment, we tested our algorithm on six meshes,[1] one of which is shown in Fig. 4. The sizes of the meshes range between 703,512 and 10,000,000 faces. All these meshes have only one connected component, except *Blade* which has 295 connected components because of many small isolated closed meshes within the blade.

Our experiments are performed with various values of ϵ. For each mesh, we computed the height of the mesh α (i.e. the maximal height minus the minimal one), and let $\epsilon = (\Phi \cdot \alpha)/100$, Φ being a percentage ($0 \leq \Phi \leq 100$). We start with $\Phi = 0.32$, and ran five experiments where, with each additional experiment, Φ is doubled. We also computed the persistence diagrams corresponding to the lower-star filtration induced by the height value on the vertices of the original mesh (without simplification). Note that as Φ increases, ϵ increases too. The average number of faces in a single cluster increases and thus the combinatorial map becomes more and more simplified. This tends to correspond to a rise in the bottleneck distance between the persistence diagrams corresponding to the simplified mesh and the original mesh, displayed in Table 1(a).

We can see an illustration of the effect of the ϵ parameter on the size of the different clusters in Fig. 5. Data showing the number of cells in the simplified mesh as Φ varies is displayed in Fig. 6(a). As expected, increasing ϵ greatly affects mesh size.

The effect of ϵ on the computation time is illustrated in Fig. 7(a), where our method for computing persistence based on 2-map simplification is ran with different values of Φ. As expected, computation time decreases while ϵ increases, as this corresponds to a simpler mesh and thus a simpler filtration. We can notice that for sufficiently large ϵ, computation time does not decrease because all the time is taken by the computation of the filtration and the simplification step.

The bottleneck distances between the persistence diagrams corresponding to the lower-star filtration and the filtration obtained when varying ϵ are given in Table 1(a). As expected, the bottleneck distances are always smaller than ϵ.

5.2. 2D Images

For our second experiment, we used six grayscale images[2] from Unsplash (https://unsplash.com/), one depicted in Fig. 8, with size between 1280 × 853 and 5616 × 3744 pixels. For each image, we created a 2D mesh having one vertex per pixel, its height value being the intensity of the pixel. This transformation enables application of our algorithm for approximating the persistent homology of 2D meshes to grayscale images. The number of cells in the simplified meshes obtained from the images are given in Fig. 8.

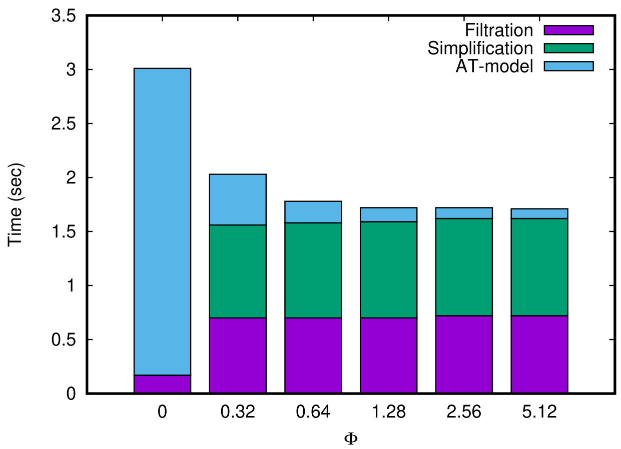
We compared the persistent homology computation of the six meshes for increasing values of Φ. The effect of ϵ on the number of cells is illustrated in Fig. 6(b).

Like for experiments with meshes, we can observe in Fig. 7(b) that computation time decreases while ϵ increases. We obtain here a better speed-up compared to the experiment with meshes. This comes from the numbers of cells which is much larger for images than for meshes. In this case, simplifying the 2-map greatly reduces the numbers of cells and thus the computation time spent in the AT-model computation.

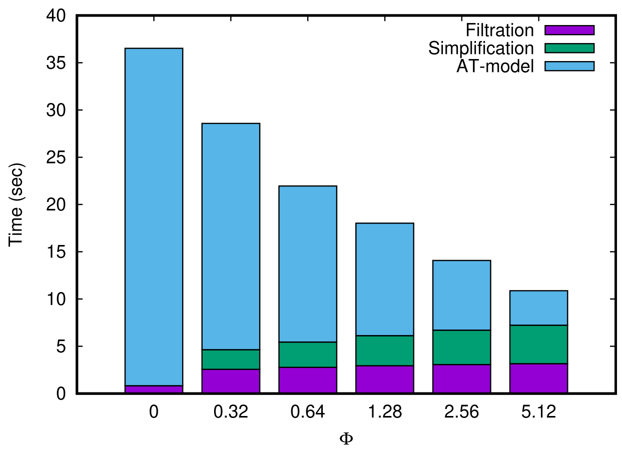
Lastly, the bottleneck distances between the persistence diagrams corresponding to the lower-star filtration and the filtration obtained for the different Φ are given in Table 1(b). As expected, the bottleneck distances are always smaller than ϵ.

[1] http://www-graphics.stanford.edu/data/3Dscanrep/ https://www.cc.gatech.edu/projects/large_models/

[2] Images taken by C. Hu, D. Dobrila, J. Alexander, K. Toth, M. Asthoff and T. Naccarato. Im1 is changyu-hu-4672-unsplash.jpg; Im2 is damjan-dobrila-39542-unsplash.jpg; Im3 is jack-alexander-131176-unsplash.jpg; Im4 is kathy-toth-31945-unsplash.jpg; Im5 is mark-asthoff-78328-unsplash.jpg and Im6 is tony-naccarato-55-unsplash.jpg

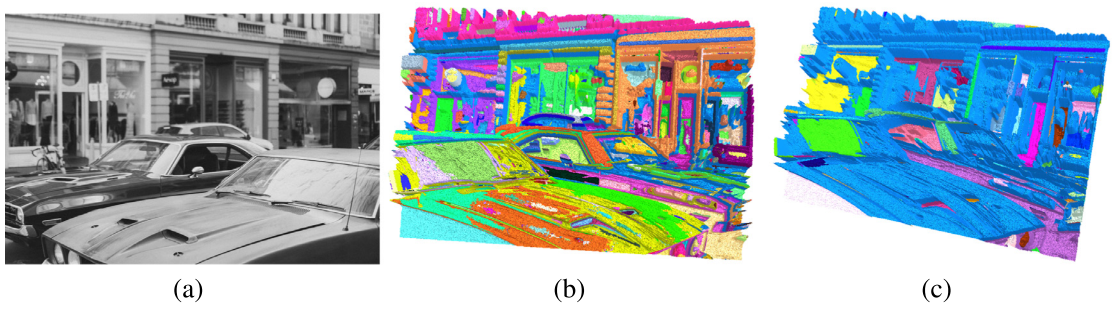


(a)



(b)

Fig. 7. Computation time (in seconds) of our method by using the lower-star filtration with increasing Φ values. For $\Phi = 0$, lower-star filtration is used without simplification step. The graph shows average values for the six meshes in (a), and for the six images in (b), and details time spent in the different parts of the method: computation of the filtration, combinatorial map simplification and persistence computation by using AT-model. `Filtration` computes the height of each cell (vertices, edges and face) and dispatches faces into clusters according to Φ. `Simplification` is the 2-map simplification, and `AT-model` is the persistent homology computation, including sorting the cells.



	Size	#0-cells	#1-cells	#2-cells
Im1	1834×2000	3,668,000	10,996,333	7,328,334
Im2	2200×1313	2,888,600	8,658,775	5,770,176
Im3	3000×2000	6,000,000	17,990,001	11,990,002
Im4	3456×5184	17,915,904	53,730,433	35,814,530
Im5	5616×3744	21,026,304	63,060,193	42,033,890
Im6	1280×853	1,091,840	3,271,255	2,179,416

Fig. 8. (a) One image used in our second experiments. (b) Its representation as mesh, with one color per cluster, for $\epsilon = 160$, (c) and for $\epsilon = 320$. The table gives the size of the images (in pixels), the number of i-cells for the six meshes built from the images, #i-cells. The number of Hi generators is the same for all images, H0 $= 1$, H1 $= 0$ and H2 $= 0$.

6. Conclusion

We conclude with a brief discussion of future directions for research. First, we plan to extend our work to non-orientable manifolds by using the generalized maps package (the non-orientable extension of combinatorial maps) of CGAL. We also would like to define a parallel version of our method: the combinatorial map simplification was already defined in parallel in [10] but we need now to study if it is possible to parallelize some parts of the AT-model computation algorithm. Extension in nD could be given based on the theoretical results for removal and contraction operations in any dimension given in [21]. Indeed, all basic tools used in this work, combinatorial maps, removal / contraction operations and AT-model computation, are defined in any dimension.

Declaration of Competing Interest

None.

References

[1] T.K. Dey, H. Edelsbrunner, S. Guha, Computational topology, in: Advances in Discrete and Computational Geometry, American Mathematical Society, 1999, pp. 109–143.
[2] M. W. Bern, D. Eppstein, P. K. Agarwal, N. Amenta, L. P. Chew, T. K. Dey, D. P. Dobkin, H. Edelsbrunner, C. Grimm, L. J. Guibas, J. Harer, J. Hass, A. Hicks, C. K. Johnson, G. Lerman, D. Letscher, P. E. Plassmann, E. Sedgwick, J. Snoeyink, J. Weeks, C. Yap, D. Zorin, Emerging challenges in computational topology, arXiv:9909001. (1999).
[3] G. Carlsson, Topology and data, Bull. Am. Math. Soc. 46 (2) (1999) 255–308.
[4] H. Edelsbrunner, J. Harer, Computational Topology - an Introduction, American Mathematical Society, 2010.
[5] D. Günther, J. Reininghaus, H. Wagner, I. Hotz, Efficient computation of 3D Morse–Smale complexes and persistent homology using discrete Morse theory, Vis. Comput. 28 (10) (2012) 959–969.
[6] V. Robins, P. Wood, A. Sheppard, Theory and algorithms for constructing discrete morse complexes from grayscale digital images, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1646–1658.
[7] J.-D. Boissonnat, S. Pritam, D. Pareek, Strong collapse for persistence, in: 26th Annual European Symposium on Algorithms, in: Leibniz International Proceedings in Informatics (LIPIcs), 112, 2018. 67:1–67:13
[8] M.B. Botnan, G. Spreemann, Approximating persistent homology in euclidean space through collapses, Appl. Algebra Eng. Commun. Comput. 26 (1) (2015) 73–101.

[9] T.K. Dey, R. Slechta, Filtration simplification for persistent homology via edge contraction, arXiv:1810.04388. (2018).

[10] G. Damiand, R. Gonzalez-Diaz, Parallel homology computation of meshes, in: Proc. of 6th Workshop on Computational Topology in Image Context, in: Lecture Notes in Computer Science, 9667, Springer International Publishing, Marseille, France, 2016, pp. 53–64.

[11] G. Damiand, R. Gonzalez-Diaz, Persistent homology computation using combinatorial map simplification, in: Proc. of 7th International Workshop on Computational Topology in Image Context (CTIC), in: Lecture Notes in Computer Science, 11382, Springer International Publishing, Malaga, Spain, 2019, pp. 26–39.

[12] P. Lienhardt, N-dimensional generalized combinatorial maps and cellular quasi-manifolds, Int. J. Comput. Geom. Appl. 4 (3) (1994) 275–324.

[13] G. Damiand, P. Lienhardt, Combinatorial Maps: Efficient Data Structures for Computer Graphics and Image Processing, A K Peters/CRC Press, 2014.

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

[15] R. Gonzalez-Diaz, P. Real, On the cohomology of 3D digital images, Discrete Appl. Math. 147 (2-3) (2005) 245–263.

[16] R. Gonzalez-Diaz, A. Ion, M.-J. Jimenez, R. Poyatos, Incremental-decremental algorithm for computing AT-models and persistent homology, in: Proc. of International Conference Computer Analysis of Images and Patterns, Seville, Spain, 2011, pp. 286–293.

[17] G. Damiand, S. Peltier, L. Fuchs, Computing homology for surfaces with generalized maps: application to 3D images, in: Proc. of 2nd Int. Symposium on Visual Computing, in: LNCS, 4292, Springer Berlin/Heidelberg, Lake Tahoe, Nevada, USA, 2006, pp. 235–244.

[18] D. Cohen-Steiner, H. Edelsbrunner, J. Harer, Stability of persistence diagrams, Discrete Comput. Geom. 37 (1) (2007) 103–120.

[19] F. Chazal, V. de Silva, M. Glisse, S.Y. Oudot, The Structure and Stability of Persistence Modules., Springer Briefs in Mathematics, Springer, 2016.

[20] G. Damiand, Combinatorial maps, CGAL User and Reference Manual, 3.9 Edition, 2011. http://www.cgal.org/Pkg/CombinatorialMaps.

[21] G. Damiand, R. Gonzalez-Diaz, S. Peltier, Removal operations in nD generalized maps for efficient homology computation, in: Proc. of 4th Int. Workshop on Computational Topology in Image Context, in: LNCS, 7309, Springer Berlin/Heidelberg, Bertinoro, Italy, 2012, pp. 20–29.