# Persistence-sensitive simplification of functions on surfaces in linear time

Dominique Attali* Marc Glisse* Samuel Hornus† Francis Lazarus*
Dmitriy Morozov‡

July 4, 2008

## Abstract

Persistence provides a way of grading the importance of homological features in the sublevel sets of a real-valued function. Following the definition given by Edelsbrunner, Morozov and Pascucci, an $\varepsilon$-simplification of a function $f$ is a function $g$ in which the homological noise of persistence less than $\varepsilon$ has been removed. In this paper, we give an algorithm for constructing an $\varepsilon$-simplification of a function defined on a triangulated surface in linear time. Our algorithm is very simple, easy to implement and follows directly from the study of the $\varepsilon$-simplification of a function on a tree. We also show that the computation of persistence defined on a graph can be performed in linear time in a RAM model. This gives an overall algorithm in linear time for both computing and simplifying the homological noise of a function $f$ on a surface.

*Gipsa-Lab, CNRS UMR 5216, Grenoble, France. first.last@gipsa-lab.inpg.fr
†INRIA Sophia Antipolis - Méditerranée, France. samuel.hornus@sophia.inria.fr
‡Duke University. morozov@cs.duke.edu

# 1 Introduction

**Motivation.** Much of modern science and engineering is driven by data. However, measuring nature is no easy task. Experiments are always plagued by noise. More fundamentally, physical phenomena do not exist at a single scale, and information obtained in experiments about these phenomena spans all scales at once.

Recently the theory of persistence homology [8, 18] emerged as a viable method for dealing with noise and omniscalar nature of data. Instead of forcing the user to make decisions about a fixed scale or a fixed level of noise in the data, persistence quantifies features across all different scales simultaneously. It presents the measurements in a persistence diagram [3] which records each feature as a point in the extended plane. The distance of the point to the diagonal represents the prominence, or persistence, of its feature. After examining such complete picture a user is free to decide how to view the data.

However, just knowing how much noise there is in the data, and where it occurs is not enough. Before using it in computation or visualization one usually desires to rid the data of unimportant features while keeping significant information intact. This raises the natural question of persistence-sensitive simplification.

**Results and prior work.** The question of simplification is tightly embedded in the history of persistence. It was considered in the first paper to introduce persistence [8] and later reinforced by the stability result [3]. Some of the first uses of persistence in the visualization community have been for simplifying various descriptors: Morse-Smale complexes [7, 2, 12] and Reeb graphs [16, 6]. We find simplification of topological descriptors convenient for visualization, but insufficient for computational analysis of the underlying data.

In this paper we follow the formalization of the simplification question proposed by Edelsbrunner, Morozov, and Pascucci [9]. Given a real-valued function $f : \mathbb{X} \to \mathbb{R}$ with a persistence diagram $\mathrm{Dgm}(f)$, they call a generic function $g : \mathbb{X} \to \mathbb{R}$ a (strong) $\varepsilon$-simplification of $f$ if the two functions are close, $\|f - g\|_\infty \leq \varepsilon$, and its persistence diagram $\mathrm{Dgm}(g)$ consists only of those points in the diagram of $f$ that are more than $\varepsilon$ away from the diagonal. Edelsbrunner et al. [9] give a constructive proof of existence of $\varepsilon$-simplifications of functions on 2-manifolds, and show that the distance between the original function and its $\varepsilon$-simplification sometimes has to be $\varepsilon$. Their algorithm applies to functions defined on the vertices of the triangulated 2-manifold, and linearly interpolated elsewhere. It is technically involved, and difficult to implement since it requires potentially considerable subdivision of the domain.

In this paper we deliberately abstract ourselves from the piecewise linear function setting, and show how to simplify general filtrations of simplicial complexes without subdividing simplices. The abstraction turns out beneficial, and we obtain a simple algorithm that computes $\varepsilon$-simplifications in linear time. Moreover, the algorithm is output-sensitive, and by virtue of its optimality allows for an efficient extraction of a hierarchy of $\varepsilon$-simplifications, thus resolving an open question of [9]. Our general setup is not too restrictive and can be applied to the piecewise linear function simplification by taking the first barycentric subdivision of the domain as explained in [14]. Additionally, we observe that in the RAM model, persistence of filtrations of 2-manifolds can be computed in linear time, thus giving us both an optimal computation and simplification algorithms in this model.

## 2 Persistence

Persistence studies evolution of classes in sequences of vector space. Most commonly such sequences arise as we consider the homology groups of a filtration of the space by the sublevel sets of a real-valued function defined on it. Intuitively homology is a topological invariant that keeps track of components, tunnels, voids, and their high-dimensional counterparts in a topological space; we refer the reader to Munkres [15] or Hatcher [13] for a review of homology theory.

### 2.1 Pairing and $\varepsilon$-simplifications

Let $f : \mathbb{X} \to \mathbb{R}$ be a real-valued function, and denote by $\mathbb{X}_a = f^{-1}(-\infty, a]$ its sublevel set, and by $\mathsf{H}_p(\mathbb{X}_a)$ the $p$-dimensional homology group of the sublevel set. For any $a < b$ inclusion of the sublevel sets $\mathbb{X}_a \subseteq \mathbb{X}_b$ induces a map between their homology groups $i_a^b : \mathsf{H}_p(\mathbb{X}_a) \to \mathsf{H}_p(\mathbb{X}_b)$, and we obtain a sequence

$$\mathsf{H}_p(\mathbb{X}_{a_1}) \to \mathsf{H}_p(\mathbb{X}_{a_2}) \to \ldots \to \mathsf{H}_p(\mathbb{X}_{a_n})$$

where $a_i$ are those function values where $p$-dimensional homology of the sublevel set changes. According to persistence, a class $\lambda \in \mathsf{H}_p(\mathbb{X}_a)$ is born in $\mathsf{H}_p(\mathbb{X}_a)$ if $\lambda$ is not in the image of the map $i_{a'}^a$ for any $a' < a$. Class $\lambda$ dies in $\mathsf{H}(\mathbb{X}_b)$ if $i_a^b(\lambda) \in \operatorname{im} i_{a'}^b$ for some $a' < a$, but $i_a^{b'}(\lambda) \notin \operatorname{im} i_{a'}^{b'}$ for any $b' < b$. In this case, we say that $\mathsf{H}_p(\mathbb{X}_a)$ and $\mathsf{H}_p(\mathbb{X}_b)$ are paired and we record this information by adding point $(a, b)$ to the $p$-dimensional diagram of the function, $\mathrm{Dgm}_p(f)$. We add a point for every such class, and for technical reasons add every point on the diagonal with infinite multiplicity. Cohen-Steiner et al. [3] have shown that persistence diagrams are stable under small perturbations of the function.

For algorithmic purposes one replaces space $\mathbb{X}$ with its triangulation $K$, and function $f$ with a filtration of the simplices in $K$. Namely, we take an ordering $\mathcal{F} = \sigma_1, \sigma_2, \ldots, \sigma_m$ of simplices of $K$, and denote by $K_i = \bigcup_{j=1}^{i} \sigma_j$ the union of simplices up to $\sigma_i$. For all $K_i$ to be subcomplexes, a face $\sigma$ of a simplex $\tau$ must precede it in the filtration, which we denote $\sigma <_{\mathcal{F}} \tau$. We write $[\sigma, \tau]_{\mathcal{F}}$ for the sequence of simplices between $\sigma$ and $\tau$. We adopt the convention that a function $f : K \to \mathbb{R}$ is always assumed to be increasing, *i.e.* its sublevel sets are subcomplexes. A function $f : K \to \mathbb{R}$ is *compatible* with the filtration $\mathcal{F}$ if $f(\sigma) \leq f(\tau)$ whenever $\sigma <_{\mathcal{F}} \tau$. Equivalently, we will also say that $\mathcal{F}$ is an $f$-filtration.

The way in which we obtain a filtration depends on the function we study. One example common in practice is a piecewise linear function $f : |K| \to \mathbb{R}$ defined on the vertices of a triangulation $K$ and linearly interpolated on the interiors of the simplices. If we assign to each simplex the maximum value the function obtains on it, we get a piecewise constant approximation $\bar{f} : K \to \mathbb{R}$ of $f : |K| \to \mathbb{R}$ with $\bar{f}(\sigma) = \max_{x \in \sigma} f(x)$. Sorting the simplices by their $\bar{f}$-value and breaking the ties by dimension and arbitrarily in each dimension, we get a total order on the simplices which gives an $f$-filtration.

The sequence of simplices implies a filtration of subcomplexes built-up simplex by simplex, $\emptyset = K_0 \subseteq K_1 \subseteq K_2 \subseteq \ldots \subseteq K_m = K$. Applying the homology functor, the inclusions induce maps between homology groups, and we obtain a sequence

$$\mathsf{H}(K_1) \to \mathsf{H}(K_2) \to \ldots \to \mathsf{H}(K_m).$$

Applying the definition of persistence to this sequence we have a homology class born with the addition of simplex $\sigma = \sigma_i$ that dies with addition of simplex $\tau = \sigma_j$. We call $\sigma$ positive and $\tau$

negative, and say that the two simplices are paired. The *persistence* of $(\sigma, \tau)$ is $f(\tau) - f(\sigma)$. For example, for the case of 0-dimensional homology, an edge is negative if it merges two components. From the oldest vertices of each component, we pick the youngest one, and say that it is paired with the negative edge.

Some positive simplices remain unpaired because the classes they create never die, we call such simplices *essential*. The number of essential $p$-simplices is equal to the rank of the $p$-dimensional homology group of $K$. Edelsbrunner, Letscher, and Zomorodian [8] gave an algorithm to compute the pairing of simplices in the filtration of a simplicial complex.

In this paper, we want to simplify a function defined on a simplicial complex in a way that removes its less persistent features. Specifically, we investigate the concept of $\varepsilon$-simplification introduced in [9] that we slightly adapt to our purpose. We first define the notion of *local pair* of a filtration. A persistence pair is local if its simplices are consecutive in the filtration. It is easy to prove that two consecutive simplices in the filtration are (locally) paired if and only if the first is a face of the second.

**Definition 1** ($\varepsilon$-simplification)**.** *A* dimension $p$ $\varepsilon$-simplification *of a function* $f : K \to \mathbb{R}$ *is a function* $g : K \to \mathbb{R}$ *such that:*

1. $\|f - g\|_\infty \leq \varepsilon$,
2. *all persistence diagrams of* $g$ *are the same as those of* $f$ *except for* $\mathrm{Dgm}_p(g)$ *which is the same as* $\mathrm{Dgm}_p(f)$ *but with all off-diagonal points at* $L_1$*-distance at most* $\varepsilon$ *from the diagonal removed,*
3. $K$ *has a* $g$*-filtration such that all pairs of persistence* 0 *are local.*

An $\varepsilon$-simplification of $f$ is defined in the same way as a dimension $p$ $\varepsilon$-simplification except that Condition 2 above is replaced by requiring that all the persistence diagrams are simplified.

## 2.2 Computing the persistence of a filtered graph in linear time

Having defined persistence for simplicial complexes, we now turn our attention to the computation of persistence for connected graphs. For this, consider a filtration of a connected graph $G = (V, E)$. Computing the persistence of this filtration amounts (1) to distinguish positive from negative edges and (2) to pair the negative edges with the (positive) vertices. As noticed in [1] for the special case of height functions over terrain meshes, (1) and (2) can be implemented using the union-find data-structure, keeping track of the oldest vertex in each component. In particular, $G$ can be interpreted as a union graph [1] with edge weights corresponding to the filtration ordering. From Kruskal's algorithm [17], we know that the set of negative edges of $G$'s filtration corresponds to the minimum spanning tree of this union graph. To obtain a linear time algorithm we use a two step algorithm:

1. We compute the minimum spanning tree $T$ of $G$ with filtration ordering for the edges.
2. We compute the persistence pairing in $T$ using $T$ itself as a batched (or off-line) union tree.

The first step can be implemented with the Fredman and Willard [10] linear time algorithm for the minimum spanning tree problem. The model of computation is a Random Access Machine with fixed word size, allowing usual arithmetic operations and bitwise Boolean operations in constant time per word. The word size is also assumed to be large enough to hold the various parameters of

the problem (when the filtration is deduced from a function on $V \cup E$, each function value should fit into a single machine word). We refer to this model simply as a RAM. The second step can be implemented in the same model of computation with the algorithm by Gabow and Tarjan [11] for union trees, see Appendix A for details. In conclusion,

**Theorem 2.** *The persistent homology of a connected graph with $m$ edges can be computed in $O(m)$-time on a RAM.*

Notice that when the filtration of $G$ is implicitly given by a function on $V \cup E$, the filtration order of the edges in $T$ can be obtained in linear time with Radix sort.

### 2.3 Persistence on $d$-manifolds via duality

In the special case where the underlying space of the simplicial complex is a $d$-manifold, an interesting notion is that of duality. [4] contains an advanced study of persistence and duality. We present here a simpler version which is sufficient for our needs. This simple result can also be deduced by applying the symmetry theorem of [4] to the first barycentric subdivision of the complex (which is the same as the first barycentric subdivision of the dual complex).

The complex is defined by the adjacency relation between its simplices. Using the same simplices and inverting the adjacency relation (*i.e.* a simplex $\tau^*$ is a face of $\sigma^*$ in the dual if and only if $\sigma$ is a face of $\tau$ in the original complex), we can define a dual complex (note that this is a polyhedral complex for 2- or 3-manifolds but is only described as a *block* complex in the general case [15]). In this paper, we will only be interested in the dual graph induced by the simplices of dimension $d$ (vertices of the graph) and $d-1$ (edges of the graph).

Persistence is defined with respect to an ordering of the simplices. When we exchange faces and cofaces, we cannot use the same ordering anymore, or cofaces would appear before faces. The ordering of simplices we use for the dual is thus simply the reverse of the original ordering. If we look at the adjacency matrix with the rows and columns in the order given by the filtration, exchanging faces and cofaces means transposing this matrix, while changing the filtration order means reversing the order of rows, and reversing the order of columns. In the end, if the original adjacency matrix is $D$, the dual adjacency matrix $D^*$ is $D$ transposed with respect to the non-standard diagonal.

A result in [5] shows that the pairing function $r_D(i,j)$ defined by 1 if $j$ kills a cycle created by $i$ and 0 otherwise can be expressed as

$$r_D(i,j) = \mathtt{rank} D_i^j - \mathtt{rank} D_{i+1}^j + \mathtt{rank} D_{i+1}^{j-1} - \mathtt{rank} D_i^{j-1},$$

where $D_i^j$ is the lower left minor of $D$ obtained by deleting the first $i-1$ rows and the last $n-j$ columns (the simplices are indexed by $1 \dots n$ here). It is straightforward to see that $r_{D^*}(n+1-j, n+1-i) = r_D(i,j)$[1]. This means that the persistence pairing is preserved by duality. In particular, if $(\sigma, \tau)$ is a pair in the primal, $(\tau^*, \sigma^*)$ is a pair in the dual, and if $\sigma$ creates an essential cycle in the primal $\sigma^*$ also creates an essential cycle in the dual. The dual of a negative simplex is positive, and the dual of a positive simplex is either negative if it is paired or positive if it creates an essential cycle.

In this paper, we focus on the simplification of the 0-homology of a graph. By duality, this directly gives a way to simplify the 1-homology of a 2-manifold, and more generally the $d-1$-homology of a $d$-manifold.

[1] Note that $i$ and $j$ are indices in the filtration, so the simplex of index $i$ in the primal is the same as the simplex of index $n+1-i$ in the dual.

# 3 $\varepsilon$-Simplification on trees

## 3.1 Statement of result

In this section, we consider a tree $T$ with $n$ vertices, a function $f : T \to \mathbb{R}$ and a positive constant $\varepsilon$. Let $\mathcal{F}$ be a $f$-filtration of $T$ and suppose the persistence pairs in $\mathcal{F}$ have been precomputed. We state our main result:

**Theorem 3.** *Given a tree $T$ with $n$ vertices, a function $f : T \to \mathbb{R}$ and a positive constant $\varepsilon$, computing an $\varepsilon$-simplification of $f$ can be performed in $O(n)$ time using $O(n)$ storage, if we assume the persistence pairs in a $f$-filtration of $T$ have been precomputed.*

## 3.2 Algorithm

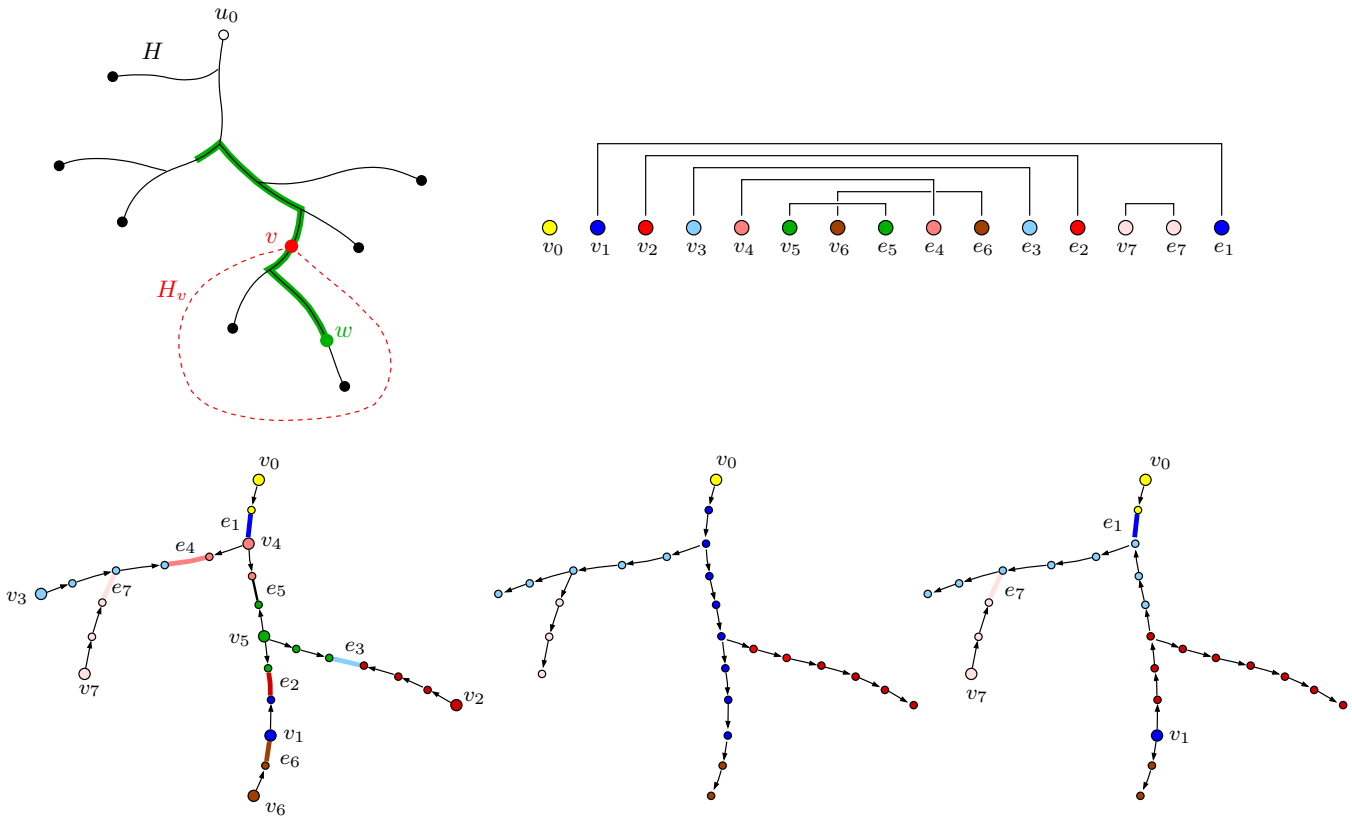


Figure 1: Top left: the connecting path starting at $w = \min_{\mathcal{F}} H_v$ passes through $v$. Top right: filtration of the tree on the bottom left. Oriented edges are locally paired with their head. Bottom middle: $(+\infty)$-simplification. Bottom right: all pairs in $[v_2, e_2]$ have been reduced.

In this section, we describe our algorithm for computing an $\varepsilon$-simplification $g$ of $f$ before proving its correctness in Section 3.3.

### 3.2.1 Special case: $\varepsilon = +\infty$

In a $f$-filtration $\mathcal{F}$ of a tree $T$, all vertices are positive and all edges are negative. The vertex $v_0$ with smallest $f$-value creates an essential connected component and is the only vertex which is unpaired. Suppose all vertices in $\mathcal{F}$ are locally paired but $v_0$. We observe that every edge must be paired with its endpoint of largest $f$-value. Let us orient the edges in $T$ from their endpoint of smallest $f$-value to their endpoint of largest $f$-value as in Figure 1 (middle). The previous observation shows that every vertex but $v_0$ has in-degree 1. In particular, $f$ is increasing on the vertices going down the tree from $v_0$.

If the pairs are not already local, an $(+\infty)$-simplification $g$ of $f$ can be constructed as follows. We define the vertex $v_0$ (which has the smallest $f$-value) as the root of $T$ and let $T_v$ be the subtree of $T$ rooted at $v$ (see Figure 1). We set $g(v_0) = f(v_0)$ and for every vertex $v \neq v_0$ we set $g(v) = \min f(T_v)$. This minimum can be computed in linear time for all the vertices by recursively removing the leaves of $T$ after initializing $g$ to $f$: if $v$ is a leaf and $w$ is its neighbor, simply set $g(w) = min(g(w), g(v))$. Once the $g$-values at the vertices have been determined as described above, we can deduce the $g$-value at the edges from their endpoints by setting $g(ab) = \max\{g(a), g(b)\}$. Furthermore, we can also construct a filtration $\mathcal{G}$ compatible with $g$ with the property that all simplices but $v_0$ are locally paired. For this, we rank simplices in such a way that every edge follows immediately its endpoint of highest rank.

### 3.2.2 General case

To construct an $\varepsilon$-simplification $g$ of $f$, we first initialize $g$ to $f$ and mark simplices that belong to pairs of persistence less than $\varepsilon$. We will prove in the next section that the set of marked simplices form a forest, in which each tree $H$ has a missing vertex $u_0$. We use this missing vertex $u_0$ as the root of the tree $\bar{H} = H \cup \{u_0\}$ and apply the simplification described above. By processing the forest componentwise as described above, this gives a linear algorithm for computing an $\varepsilon$-simplification $g$ of $f$. The data-structure is reduced to $T$ and has linear size. The correctness of our algorithm will be established in the following section.

Note that the set of simplices in $H$ which are assigned the same value $f(w)$ forms a subpath of the path from $w$ to $u_0$. This is a direct consequence of the fact that $H_v \subseteq H_w$ iff $w$ is an ancestor of $v$ in the tree $\bar{H}$ rooted at $u_0$. This subpath is closed at $w$ and open at the other extremity and we call it the *flattening path* of $w$.

Let $P$ be the set of simplices belonging to pairs of persistence less than $\varepsilon$. Instrumental to our proof, we construct a filtration $\mathcal{G}$ compatible with $g$ with the property that all simplices in $P$ are locally paired in $\mathcal{G}$ and the pairing and relative order of simplices not in $P$ are the same in $\mathcal{F}$ and $\mathcal{G}$. For this, we reorder simplices in $\mathcal{F}$ by moving all simplices on the flattening path of $w$ next to $w$ and reordering them in such a way that they are locally paired in $\mathcal{G}$.

## 3.3 Proof of correctness

We say that a set of persistence pairs $\mathcal{P}$ is *closed under inclusion* if whenever $(\sigma, \tau) \in \mathcal{P}$ and $(\sigma', \tau')$ is a persistence pair with $[\sigma', \tau']_{\mathcal{F}} \subseteq [\sigma, \tau]_{\mathcal{F}}$, then $(\sigma', \tau') \in \mathcal{P}$. In particular, the set of pairs with persistence less than $\varepsilon$ is closed under inclusion. Given a set of persistence pairs $\mathcal{P}$ closed under inclusion, we first study the way simplices in $\mathcal{P}$ are arranged in the tree $T$. We then define connecting paths and $\mathcal{P}$-reductions which will be instrumental in establishing the correctness of our algorithm.

We start by characterizing $(0,1)$-pairs. Let $(v,e)$ be a persistence pair in $\mathcal{F}$. Adding $e$ to the filtration $\mathcal{F}$ merges two components $C_v$ and $C_w$. Moreover, $v = \min_{\mathcal{F}} C_v$ and $w = \min_{\mathcal{F}} C_w$ for some vertex $w$ which precedes $v$ in the filtration, $v >_{\mathcal{F}} w$. Let $p(v,e)$ be the half-open path in the tree $T$ connecting $v$ to $e$. Clearly,

$$p(v,e) \;\subseteq\; C_v \cup \{e\} \;\subseteq\; [v,e]_{\mathcal{F}}. \tag{1}$$

**Lemma 4.** *Any simplex $\sigma \in p(v,e)$ is paired with some simplex $\tau \in [v,e]_{\mathcal{F}}$. In particular, $[\sigma,\tau]_{\mathcal{F}} \subseteq [v,e]_{\mathcal{F}}$.*

*Proof.* Let $\sigma \in p(v,e)$. We may assume $\sigma \neq v, e$. By Equation (1), we have $\sigma \in C_v$. Suppose $\sigma$ is a vertex and let $e(\sigma)$ be its paired edge in $\mathcal{F}$. When $e(\sigma)$ is added it merges two components one of whom contains $\sigma$ as minimal vertex. So the component of $\sigma$ at that time must be strictly included in $C_v$, hence $e(\sigma) <_{\mathcal{F}} e$. Suppose now that $\sigma$ is an edge and denote $v(\sigma)$ its paired vertex. As $\sigma \in C_v$, the two components it joins when $\sigma$ is added must be included in $C_v$, hence $v(\sigma) \in C_v$ and $v(\sigma) >_{\mathcal{F}} v$. □

Let us denote by $\{\mathcal{P}\}$ the set of simplices in the pairs of $\mathcal{P}$.

**Lemma 5.** *Consider a connected component $H$ of $\{\mathcal{P}\}$ in $T$. Then, any simplex in $H$ is paired with a simplex in $H$.*

*Proof.* Let $(v,e)$ be a pair in $\mathcal{P}$. Since $\mathcal{P}$ is closed by inclusion, Lemma 4 shows that $p(v,e) \subseteq \{\mathcal{P}\}$. It follows that $e$ and $v$ belong to a same component of $\{\mathcal{P}\}$ in $T$. □

Let $\bar{H}$ be the closure of a component $H$ of $\{\mathcal{P}\}$ in $T$. It follows from Lemma 5 that each component $H$ of $\{\mathcal{P}\}$ is a subtree of $T$ which contains the same number of vertices and edges. Therefore, $\bar{H} \setminus H$ must be reduced to a single vertex which we call the root of $H$. The next lemma says that this root must be lower than any simplex in $H$:

**Lemma 6.** *The root of $H$ is lower in $\mathcal{F}$ than any simplex in $H$.*

*Proof.* Let $u_0$ be the root of $H$. We show that the hypothesis $u_0 >_{\mathcal{F}} \min_{\mathcal{F}} H$ leads to a contradiction. Consider the lowest (w.r.t. $\mathcal{F}$) edge $e \in H$ that connects $u_0$ to a vertex of $H$ lower than $u_0$. We claim that $e$ must be paired with a vertex $v \in H$ lower than $u_0$. Indeed, when $e$ is added in the filtration, $e$ connects two components: one of whom contains $u_0$ and possibly vertices of $H$ higher than $u_0$, while the other component contains at least one vertex of $H$ lower than $u_0$. Since $e$ is paired in $H$, the claim follows. Let $e(u_0)$ be paired with $u_0$. By the assumption on $\mathcal{P}$ we must have $e(u_0) >_{\mathcal{F}} e$. But this contradicts that $e(u_0)$ is paired with $u_0$ since the component below $e(u_0)$ that contains $u_0$ also contains $v$ (they are connected by a path below $e$) which is lower than $u_0$ by the above hypothesis. □

Using Lemma 6, we now define connecting paths which will turn out to be useful objects to bound the difference between $f$ and $g$ at the end of this section. Consider a non-essential vertex $v$ and the edge $e$ paired to $v$. Let $\text{Closure}_{\mathcal{F}}(v,e)$ be the set of persistence pairs obtained by taking the closure of $(v,e)$ under inclusion in $\mathcal{F}$. Applying what we just saw, $\text{Closure}_{\mathcal{F}}(v,e)$ is a forest, the roots of which we just defined.

**Definition 7.** *The* connecting path *of $(v,e)$ is the half-open path connecting $v$ to the root of the tree containing $v$ in $\{\text{Closure}_{\mathcal{F}}(v,e)\}$.*

The connecting path of $(v, e)$ is closed at $v$ and open at the other extremity $w$ and $w <_{\mathcal{F}} v$. It follows immediately from the definition that the connecting path is contained in $[v, e]_{\mathcal{F}}$.

For simplicity, we call $\mathcal{P}$-reduction of a $f$-filtration $\mathcal{F}$ the filtration $\mathcal{G}$ and its associated function $g$ constructed in Section 3.2.2. Some key properties are that $<_{\mathcal{F}}$ and $<_{\mathcal{G}}$ agree on $K \setminus \{\mathcal{P}\}$, simplices of $\{\mathcal{P}\}$ can only be lowered from $\mathcal{F}$ to $\mathcal{G}$ and $\mathcal{P}$ is composed only of local pairs in $\mathcal{G}$.

**Lemma 8.** *Given a filtration $\mathcal{F}$ and a $\mathcal{P}$-reduction $\mathcal{G}$, the pairing of simplices not in $\{\mathcal{P}\}$ is the same in $\mathcal{F}$ and $\mathcal{G}$.*

A proof is provided in Appendix B.

**Proof of Theorem 3** We are now ready to prove that the function $g$ computed by the algorithm in Section 3.2.2 is an $\varepsilon$-simplification of $f$.

*Proof.* Let $\mathcal{P}$ be the set of $(0, 1)$-pairs with persistence less than $\varepsilon$. Consider the $\mathcal{P}$-reduction $\mathcal{G}$ of $\mathcal{F}$ constructed in Section 3.2.2. By Lemma 8, $\mathcal{F}$ and $\mathcal{G}$ have the same set of $(0, 1)$-pairs with persistence higher than $\varepsilon$. Other persistence pairs of $\mathcal{G}$ are local and have zero persistence. Therefore, the persistence diagram of $g$ is the same as the persistence diagram of $f$ but with all off-diagonal points at $L_1$-distance at most $\varepsilon$ from the diagonal removed. To complete the proof, we now need to establish that $\|f - g\| \leq \varepsilon$. For this, consider a component $H$ of $\{\mathcal{P}\}$ in $T$ and let $u_0 = \bar{H} \setminus H$ be its missing vertex. Consider also a vertex $v \in H$. Let $H_v$ be the subtree of $\bar{H}$ rooted at $v$ and let $w = \min_{\mathcal{F}} H_v$, i.e. $g(v) = f(w)$. The function $g$ will be an $\varepsilon$-simplification if we can prove that $f(w) \leq f(v) \leq f(e(w))$ where $e(w)$ is the edge paired with $w$ in $\mathcal{F}$. In this abstract, we will omit the proof of the similar statement for edges in $H$. By definition, the connecting path of $(w, e(w))$ is contained in $\text{Closure}_{\mathcal{F}}(w, e(w)) \subseteq \mathcal{P}$ and therefore is contained in $H$. It starts from $w$ and ends at a vertex whose $f$-value is smaller than the $f$-value of $w$ while staying inside $H$. On the other hand $w$ is the simplex of $H_v$ with smallest $f$-value. This shows that the connecting path of $(w, e(w))$ has to go outside $H_v$, passing through $v$ (see Figure 1). Thus, $v$ belongs to the connecting path of $(w, e(w))$ which is contained in $[w, e(w)]_{\mathcal{F}}$. It follows that $f(w) \leq f(v) \leq f(e(w))$ and since $(w, e(w))$ has persistence less than $\varepsilon$, $\|f(v) - g(v)\| \leq \varepsilon$. □

# 4 Consequences and limitations

## 4.1 Consequences for the 0-homology of a complex.

The goal of this section is to prove that in a complex, we can simplify the dimension 0 persistent homology using our algorithm without affecting or being affected by higher dimensions.

The standard algorithm to compute persistence, described for instance in [5], is a special case of Gauss reduction on the adjacency matrix. To compute the dimension $k$ persistent homology, it starts with the matrix that represents the boundary operator restricted to chains of dimension $k+1$ (its image is in the chains of dimension $k$), where lines and columns are sorted according to the filtration. It then performs column additions to reduce the matrix. For each column starting from the left-most one, it looks at its lowest non-zero entry (*i.e.* a one) and adds previous columns until the lowest one is as high as it can. More precisely, as long as it can find a previous column that has its lowest one in the same row as this column, it adds that column. In the end, columns with only zeros correspond to positive simplices, and for other columns the lowest one (which is unique to this column) corresponds to the simplex paired with the simplex of this column.

Notice that at any time during this algorithm, a column, being a sum of boundaries, represents a cycle. Thus the row of the lowest one of a non-zero column always corresponds to a positive $k$-simplex (the last simplex of a cycle creates this cycle). We can then notice that rows that correspond to negative $k$-simplices play absolutely no role in this algorithm. Indeed, if we remove the lines that correspond to negative $k$-simplices and apply the same algorithm as before, we will perform exactly the same column reductions and obtain the same pairing of simplices. Moreover, if the list of positive $k+1$ simplices is known in advance, they can also be ignored in the algorithm as their column becomes null and does not play any role. These remarks prove the following lemma:

**Lemma 9.** *The pairing of the dimension $k$ persistent homology depends only on the relative order of the positive $k$-simplices and the relative order of the negative $k+1$-simplices.*

Let $\mathcal{F}$ be a filtration of a connected simplicial complex $K$. The vertices and negative edges of $\mathcal{F}$ form a tree. The simplification algorithm of Section 3 only lowers the value of the simplices of this tree. The positive edges of $\mathcal{F}$ thus remain positive after the simplification. Indeed, when a positive edge appears in $\mathcal{F}$, its extremities are already in the same connected component, and this remains true if we only add vertices and edges before this edge. The negative edges therefore also remain negative (their number does not depend on $\mathcal{F}$). The tree of negative edges thus remains the same during the algorithm. Using Lemma 9, this proves that our algorithm is not affected by the presence of positive edges and higher dimensional simplices. It also proves that persistent homology of dimension 1 or higher is not affected by our algorithm.

### 4.2 Consequences for surfaces.

As stated in the title of this paper, our goal is to simplify a function on a surface. That is, our main interest is in a complex whose underlying space is a 2-manifold.

Section 4.1 proves that we can find an $\varepsilon$-simplification of the dimension 0 persistent homology of a function on such a complex without affecting the dimension 1 persistence.

Using the result of Section 2.3, we know that the dimension 1 persistence is simply the dimension 0 persistence of a dual graph. We can therefore apply the algorithm of Section 3 in the dual. Besides, it is straightforward to see that the last paragraph of Section 4.1 also applies for this dual simplification, *i.e.* that since the dual simplification only moves positive edges and negative triangles higher, simplices don't change sign and the algorithm does not affect and is not affected by simplices which are not positive edges or negative triangles. We therefore obtain an $\varepsilon$-simplification of the dimension 1 persistence without affecting the dimension 0 persistence.

The set of simplices whose value can be changed by the dimension 0 simplification and the dimension 1 simplification are disjoint (they don't have the same dimension and sign). Since each simplification changes the value of a simplex by at most $\varepsilon$, cumulating the two simplifications also changes the value of each simplex by at most $\varepsilon$. We thus have the following theorem:

**Theorem 10.** *Given a function $f$ on a triangulated 2-manifold $K$, there exists an $\varepsilon$-simplification $g$ of $f$. Moreover, given the persistence pairing of a $f$-filtration of $K$, we can compute an $\varepsilon$-simplification $g$ and the $g$-filtration in linear time.*

In conjunction with Theorem 2, this gives:

**Corollary 11.** *In the special case of the RAM model, given only a function $f$, an $\varepsilon$-simplification of $f$ can be computed in linear time.*

If the underlying space of $K$ is a $d$-manifold with $d > 2$, the algorithm presented for 2-manifolds can be adapted to compute a function $g$ that is $\varepsilon$-close to $f$ and has simplified persistent homologies of dimension 0 and $d-1$. The intermediate persistent homologies are not affected. This is very similar to a classical result in Morse theory where, after canceling handles (by inverting the gradient between two critical points), a connected $d$-manifold has only one critical point of dimension 0 and one critical point of dimension $d$ left. Our result is a combinatorial and hierarchical (we can do partial simplification) version with bounds on the modification of the function.

### 4.3 Optimal filtrations and counterexamples

As $\varepsilon$-simplifications can always be performed on surfaces, a natural question is to extend the process to higher dimensional triangulated manifolds. We show that, in some restricted sense of optimal simplification, this is not always possible already for 3-manifolds and general complexes of dimension 2. In the following we only consider filtrations where simplices are added one at a time. We assume the reader familiar with some basics of algebraic topology and refer to the textbook by A. Hatcher [13] for the various definitions.

**Definition 12.** *Let $K$ be a triangulated manifold. A filtration of $K$ is said* optimal *if all its non-essential simplices are locally paired. In other words, an optimal simplification is an $\varepsilon$-simplification with $\varepsilon = +\infty$.*

If $\sigma$ and $\tau$ are two simplices of $K$ such that $\sigma$ is a face of $\tau$ but is not a face of any other simplex in $K$, then we say that $K$ *collapses* to $K - \{\sigma, \tau\}$. Furthermore, $K$ is said *collapsible* if it can be reduced to a vertex by a sequence collapses. It can be proved that a collapsible complex is simply connected, i.e. has a trivial fundamental group. The following lemma is easy.

**Lemma 13.** *Let $\sigma_1, \sigma_2, \ldots, \sigma_n$ be the ordering of the simplices of a filtration of $K$. If $(\sigma_i, \sigma_{i+1})$ is a local pair, then $K_{i+1}$ collapses to $K_{i-1}$, where $K_j$ denotes the subcomplex of $K$ spanned by the $j$ first simplices.*

**Claim 14.** *A triangulated Poincaré homology 3-sphere has no optimal filtration.*

*Proof.* Let $\mathcal{K}$ be a filtration of a triangulated homology 3-sphere $K$. Since a homology sphere has the homology of a sphere, $\mathcal{K}$ has precisely two essential simplices: a vertex $\sigma$ and a tetrahedron $\tau$. Clearly, from Section 2.1, $\sigma$ must be the first simplex in the filtration. Also, since the 3-cycle space of a proper subcomplex of a connected 3-manifold is trivial, the only positive – hence essential – tetrahedron must be the last simplex in $\mathcal{K}$. If $\mathcal{K}$ was optimal, then by the preceding Lemma, $K - \tau$ would be collapsible, hence simply connected. Since the fundamental group of $K$ only depends on its 2-skeleton, $K$ would also be simply connected. A contradiction. □

In fact, 2-complexes may already lack an optimal filtration. This will be the case for non-collapsible contractible complexes, as is easily shown with the same type of arguments as above. In particular,

**Claim 15.** *A triangulated dunce cap or Bing's house with two rooms has no optimal filtration.*

## References


[1] Pankaj K. Agarwal, Lars Arge, and Ke Yi. I/O-efficient batched union-find and its applications to terrain analysis. In *twenty-second annual symposium on Computational geometry*, pages 167 – 176. ACM, 2006.

[2] Peer-Timo Bremer, Herbert Edelsbrunner, Bernd Hamann, and Valerio Pascucci. A topological hierarchy for functions on triangulated surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 10(4):385–396, 2004.

[3] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of Persistence Diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007.

[4] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundations of Computational Mathematics*, 2008 (to appear).

[5] David Cohen-Steiner, Herbert Edelsbrunner, and Dmitriy Morozov. Vines and vineyards by updating persistence in linear time. In *SCG '06: Proceedings of the twenty-second annual symposium on Computational geometry*, pages 119–126, New York, NY, USA, 2006. ACM.

[6] Kree Cole-McLaughlin, Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Loops in Reeb graphs of 2-manifolds. In *Proc. 19th Ann. Sympos. Comput. Geom.*, pages 344–350, 2003.

[7] Herbert Edelsbrunner, John Harer, and Afra Zomorodian. Hierarchical morse-smale complexes for piecewise linear 2-manifolds. *Discrete and Computational Geometry*, 30:87–107, 2003.

[8] Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. Topological Persistence and Simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

[9] Herbert Edelsbrunner, Dmitriy Morozov, and Valerio Pascucci. Persistence-Sensitive Simplification of Functions on 2-Manifolds. In *22nd Annual ACM Symposium on Computational Geometry*, pages 127–134, 2006.

[10] Michael L. Fredman and Dan E. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Journal of Computer System Sciences*, 48(3):533–551, 1994.

[11] Harold N. Gabow and Robert E. Tarjan. A linear-time algorithm for a special case of disjoint set union. *Journal of Computer and System Sciences*, 30(2):209–221, 1985.

[12] Attila Gyulassy, Vijay Natarajan, Valerio Pascucci, Peer-Timo Bremer, and Bernd Hamann. Topology-based simplification for feature extraction from 3D scalar fields. In *Proceedings of the IEEE Visualization Conference*, pages 275–280, 2005.

[13] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[14] Dmitriy Morozov. *Homological Illusions of Persistence and Stability*. PhD thesis, Department of Computer Science, Duke Unversity, 2008 (to appear).

[15] James R. Munkres. *Elements of Algebraic Topology*. Addison Wesley Publishing Company, 1984.

[16] Georges Reeb. Sur les points singuliers d'une forme de Pfaff complèment intégrable ou d'une fonction numérique. *Comptes Rendus de L'Académie des Séances*, 222:847–849, 1946.

[17] Robert E. Tarjan. *Data Structures and Network Algorithms*. Number 44 in CBMS-NFS Regional conference series in applied mathematics. SIAM, 1983.

[18] Afra Zomorodian and Gunnar Carlsson. Computing Persistent Homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.

# A Computing persistence on a tree in linear time

This computation is based on the algorithm by Gabow and Tarjan [11] for union trees. We first pick an arbitrary vertex as the root of the tree $T$ and orient in linear time the edges of $T$ toward this root. Thus, each non-root vertex $v$ has a unique parent $p(v)$. We next perform a sequence of Union operations corresponding to the edges of $T$ and maintain a pointer to the oldest vertex in each component. If $e = (v, p(v))$ is a tree edge, Gabow and Tarjan define a $Union(v, p(v))$ operation that only uses $v$ as a parameter and is referred to as $Link(v)$. This leads to the following implementation of the second step, where we assume that each vertex (representative of a set) has a field *oldest* pointing to the oldest vertex in its component. Here, $Find(v)$ returns as usual the representative of the component of $v$.

```
for each edge e = (v, p(v)) of T taken in filtration order
  pvSet ← Find(p(v)); // It results from [11] that v = Find(v) at this point.
  if v.oldest is younger than pvSet.oldest then
    output (v.oldest, e) as a persistence pair;
  else
    output (pvSet.oldest, e) as a persistence pair;
    pvSet.oldest ← v.oldest;
  endif
  Link(v); // The representative of the set p(v) becomes the
           // representative for the union of the sets of v and p(v).
endfor
```

Putting $n = |V|$, this algorithm performs an intermixed sequence of $2n - 2$ Find and Link operations, which takes $O(n)$ time according to [11].

# B Proof of Lemma 8

Consider $f : \mathcal{F} \to \mathbb{R}$ and $g : \mathcal{G} \to \mathbb{R}$ such that $g$ is a $\mathcal{P}$-reduction of $f$. Consider a $(0, 1)$-pair $(v, e)$ of $\mathcal{F}$ not in $\mathcal{P}$. As above, we denote $C_v$ and $C_w$ the two components merged by $e$ when it is added in $\mathcal{F}$. Let $C'_v$ and $C'_w$ be the two components merged by $e$ in $\mathcal{G}$. Since reducing only lowers simplices of $\{\mathcal{P}\}$, $C_v$ and $C_w$ can only grow in $\mathcal{G}$, i.e $C_v \subseteq C'_v$ and $C_w \subseteq C'_w$. Suppose for the purpose of contradiction that $e$ is not paired with $v$ in $\mathcal{G}$, and that $v$ is the lowest vertex of $\mathcal{F} \setminus \{\mathcal{P}\}$ whose pairing is different in $\mathcal{G}$. Then the lowest vertex $u$ of $C'_v$ in $\mathcal{G}$ is not $v$. If $u \in C_v$, $u \in \{\mathcal{P}\}$ (its value changed). Since the simplices of $\{\mathcal{P}\}$ are paired with vertices of $\{\mathcal{P}\}$ in $\mathcal{G}$ (they form local pairs), $e$ cannot be paired with $u$, so it must be paired with the lowest vertex $w'$ of $C'_w$ which is at least as low as $w$ and then lower than $v$. For the same reason as before, $w' \notin \{\mathcal{P}\}$. $w'$ is then a vertex of $\mathcal{F} \setminus \{\mathcal{P}\}$ whose pairing is different in $\mathcal{G}$ and is lower than $v$ which contradicts our hypothesis. Assuming now that $u \in C'_v \setminus C_v$, the path from $v$ to $u$ must contain an edge of $C'_v$ whose $f$-value is larger than the $f$-value of $e$ in $\mathcal{F}$ (otherwise that path would already be in $C_v$). Consider the edge $\eta$ on this path with largest $f$-value. This edge must be an edge of $\{\mathcal{P}\}$. It merges in $\mathcal{F}$ two components, one contains $w$ and the other one contains $u$. So $\eta$ must be paired in $\mathcal{F}$ with a vertex $\nu$ lower than $u$ or $w$. But this implies $[v, e]_{\mathcal{F}} \subseteq [\nu, \eta]_{\mathcal{F}}$ and contradicts that $\mathcal{P}$ is closed by inclusion.

# C Application to the topological simplification of terrains

We wrote an implementation of our algorithms using the python language. It is specialized to the topological simplification of terrains. The terrain is a triangulated 2D grid whose simplices are assigned a height value in $[0, 1]$. The terrain is made manifold by gluing dummy triangles from the boundary of the terrain to a dummy vertex, thus forming a topological 2-sphere. The dummy simplices are assigned height greater that one. In practice, this is not ideal, as the many dummy triangles and edges tend to interfere with the pairing of the actual terrain's simplices; it would be preferable to use a single dummy 2-dimensional face whose boundary

spans the whole terrain boundary edges. Such a dummy face would become the only positive (and un-paired) 2-simplex, and would therefore not interfere with the pairing of the actual simplices. Figure 2 shows two simplifications of the height function of a terrain. Additional views of the same terrain, including 3D views, can be found at the following URL: `http://www-sop.inria.fr/members/Samuel.Hornus/simplif/`.

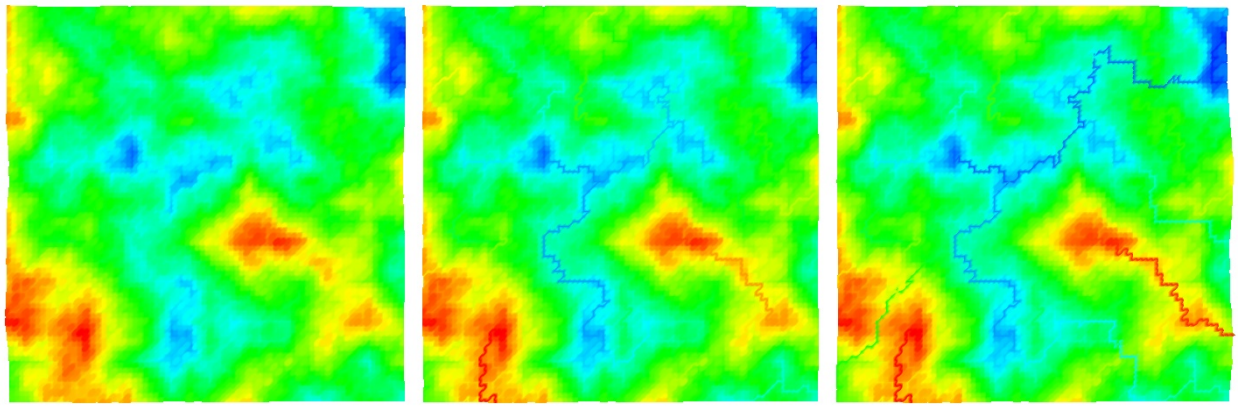

Figure 2: (This is a color figure) *Left*: a 80x80 terrain. Heights range from blue (low = 0.0), through cyan, green, yellow to red (high = 1.0). *Middle*: A 0.2-simplification of the height function has been computed. *Right*: A 1-simplification of the height function has been computed. The terrain has 38396 simplices. The computation of the spanning tree took 1.32 seconds. Each computation of the persistence pairs took roughly 2.37 seconds. Each simplification step took roughly 0.41 seconds. Timings were measured on a Core 2 duo 2.6 GHz processor.