OPTIMAL PERSISTENT HOMOLOGY PREPROCESSING USING ITERATED MORSE DECOMPOSITION

PAWEŁ DŁOTKO AND HUBERT WAGNER

ABSTRACT. In this paper we present a new preprocessing technique which simplifies a filtered complex while preserving its persistent homology. Our method is based based on discrete Morse theory. We prove that the simplification is optimal, in the sense that the resulting complex has the minimal number of cells. To achieve this, we use the concept of iterated Morse decomposition yielding a perfect iterated Morse complex. Finally, we show that using graph-theoretical algorithms our method can be implemented in an efficient way.

1. Overview

The general problem we tackle is to compute persistent homology of a filtered chain complex in any dimension. The complex can be simplicial, cubical, or more generally an abstract chain complex. We filter the complex according to the values of a function defined on its cells. Often this function is defined on 0-cells and extended to all the cells by the so-called lower star filtration. The output of the entire procedure is the list of persistence intervals.

In this paper we focus on a preprocessing step. We present an iterative simplification scheme, which results in a reduced complex, having the least possible number of cells and isomorphic persistent homology. At each iteration we use a sub-optimal simplification method. We point out that this method works for persistence and standard homology with field coefficients. In this paper we focus on persistence with $\mathbb{Z}_2$ coefficients.

2. Motivation

Our research in the direction of efficient computations of persistent homology is directly motivated by the problems we encountered in our ongoing project. In that project we aim at analyzing large, high-dimensional datasets coming from an application in text-mining. It turned out that the existing methods, listed in the next section, were inefficient when faced with such datasets [26].

We aimed at a preprocessing method, which would reduce the size of data before applying standard computations. In particular we simplify the complex while retaining essential information about persistent homology, namely features of non-zero persistence (lifespan). The method we introduce is based on discrete Morse theory (DMT). DMT is often used as a preprocessing for (standard) homology computations [18], and recently also for persistence [21]. However, directly applying this theory is quite limiting, since nothing is guaranteed about the size of the reduced complex. We strove to have a method which is provably optimal, in the sense that the simplified complex has the minimum number of cells. Doing a similar thing directly using DMT is known to be NP-hard [18].

The following paper describes the theory and algorithms for computing persistent homology using *iterated* Morse decomposition. We prove that the algorithm is correct for chain complexes of any dimension. This includes commonly used simplicial and cubical complexes. Further, we show that the algorithms can be entirely expressed in terms of basic graph-theoretical techniques. We hope that using efficient, distributed graph libraries [19] will result in a scalable implementation.

The paper is structured as follows. In Section 3 the previous work on this subject is summarized. In Section 4 the necessary background in (persistent) homology is given. In Section 5 a brief introduction to discrete Morse theory is provided. It is explained how the discrete Morse

theory can be used to compute persistence and that the Morse complex can be build by using only graph algorithms. In Section 6 the concepts of iterated Morse complex and perfect iterated Morse complex are introduced and the simplification algorithm is given. Finally in Section 7 conclusions are drawn. Additionally, we give some supplementary material in the Appendix.

3. Previous work.

In 2003, after a few earlier iterations ([27], [11], [23]), persistent homology was introduced in its contemporary form [8]. In the same paper, a matrix-reduction algorithm to compute persistence was given. This is considered the standard algorithm to compute persistence. For a comprehensive presentation the reader should consult [7]. It works for general complexes in arbitrary dimensions. The worst-case complexity is $O(n^3)$, where n is the size of the input complex. Milosavljevic et al. [20] showed that persistent homology can be computed in matrix multiplication time ($O(n^{2.3727})$ as of now). Chen and Kerber [4] proposed a randomized algorithm to compute only pairs with persistence above a chosen threshold. Despite improving the theoretical complexity, it is unclear whether these methods are better in practice. A recent variation of the standard algorithm, introduced by Chen and Kerber [5] significantly reduces the amount of computations. This idea was also used in [25] to compute persistence for $n-$dimensional images.

An earlier algorithm by Delfinado and Edelsbrunner for incremental Betti numbers computation [6] can be adapted for persistence. This algorithm exhibits linear time complexity and works for sub-triangulations of 3 dimensional sphere. When focusing on 0-dimensional homology, union-find data structures can be used to compute persistence in time $O(n\alpha(n))$ [7], which is almost linear.

Discrete Morse theory (DMT) was introduced by Robin Forman [10]. Later a more general, algebraic version was developed [17]. Using DMT for homology computations was introduced and developed by Lewiner [18]. The computational aspects of Morse complex construction were discussed in [16] and [18]. The notions of $F-$perfect and $F-$optimal Morse complexes were discussed in [2]. A simplification algorithm for a Morse complexes on 2-manifolds was been presented in [3]. A divide and conquer algorithm to compute Morse complexes was presented in [13].

Recently Abhishek Rathod and collaborators [1] reported progress on approximate algorithms for Morse matching problems. They proved approximation bounds with respect to the optimal Morse complex (but not the perfect one).

Recently Robins, Wood and Sheppard have provided a practical link between discrete Morse theory and persistence [24]. In this paper they introduced an optimal simplification scheme for persistence in case 3-dimensional complexes. This approach aims at reducing the size of the boundary matrix, while preserving persistence. In case of 3D grayscale images, a parallel implementation was reported to increase memory and runtime efficiency up to an order of magnitude [12]. The algorithm used in [24] crucially depends on simple-homotopy theory, which makes it hard to directly generalize the optimality result to higher dimensions.

A recent paper by Mischaikow et al. [21] proposes a handy theoretical framework, where discrete Morse theory is extended from complexes to filtrations. In short, by performing a Morse matching independently for each filtration level, a simplified filtered Morse complex is obtained. Both methods [24, 21] use the standard matrix-reduction algorithm as the final step of computations.

The idea of iterating Morse complex construction was proposed in [14, 21] as a heuristic to decrease the size of complexes before standard algebraic computations. In [26] it was proven that such an iteration can be used to *compute* standard homology with field coefficients. In such a case each cell of the resulting complex corresponds to exactly one homology generator of the initial complex.

Prompted by the recent contributions we extended the approach of [26] to the setting of persistent homology. In the approach presented in this paper we merge the existing techniques

to iteratively simplify the input filtered complex. In short, we build a Morse complex of a Morse complex... of the input complex, while Morse matchings are performed separately for each filtration level. We prove that the resulting *perfect iterated Morse complex*, has two properties: 1) It has the same persistent homology as the input complex. 2) It has the smallest possible number of cells. We want to point out that this generalizes the result obtained by Vanessa Robins and collaborators in [24].

4. Background

4.1. **Chain complexes and homology.** We assume that the input data is represented as a *chain complex* with field coefficients. In the most typical case this chain complex comes from a CW-decomposition of a given space. In practice, simplicial and cubical complexes are used. For simplicity, we will use $\mathbb{Z}_2$ coefficients throughout the paper, as this is the standard setting for persistence. However, we want to remark that the presented algorithms work for any field coefficients. Intuitively, in this setting a chain complex can be viewed as a set of abstract cells connected by a boundary operator as specified below.

Let us fix a chain complex $\mathcal{K}$. Let a *p-chain* be a formal sum of p-cells with the $\mathbb{Z}_2$ coefficients. The boundary operator ∂_p maps p-chains into $p-1$-dimensional chains. The definition of chain complexes requires that $\partial_p \circ \partial_{p+1} = 0$.

The chain of (co-)faces is called a (co-)boundary. For any p-chain $c = \sum a_i c_i$, we have $\partial_p c = \sum a_i \partial_p c_i$. The p-chains of $\mathcal{K}$, together with addition modulo 2, form a *group of p-chains*, denoted by $C_p(\mathcal{K})$.

The boundary operator ∂_p can be written as a binary matrix (also denoted ∂_p), whose columns represent the boundaries and rows represent coboundaries of cells.

To define homology let us first introduce the group of p-cycles, $Z_p(\mathcal{K}) = ker\partial_p$ and its subgroup: the group of p-boundaries, $B_p(\mathcal{K}) = im\partial_{p+1}$. The p-th homology group is the quotient $H_p(\mathcal{K}) = Z_p(\mathcal{K})/B_p(\mathcal{K})$. The p-th Betti number, denoted by $\beta_p(\mathcal{K})$, is the rank of this group.

4.2. **Filtrations and persistence.** Given a complex $\mathcal{K}$ and a filtering function $g : \mathcal{K} \to \mathbb{Z}$, *persistent* homology studies homological changes of the sub-level complexes, $\mathcal{K}_t = g^{-1}(-\infty, t]$. We require that $g(a) \leq g(B)$ whenever a is a face of B. Persistent homology captures the birth and death times of homology classes of the sub-level complexes, as t grows from $-\infty$ to $+\infty$. By birth, we mean that a homology feature is created; by death, we mean it either becomes trivial or becomes identical to some other class born earlier. The *persistence*, or lifetime of a class, is the difference between the death and birth times. Often a multiset of *persistence intervals* is used to represent persistence in a given dimension. A single interval encodes a lifetime of a homology class. We say that two spaces have same persistence, if their persistence intervals are the same in the corresponding dimensions. We disregard zero-length persistence intervals as nonessential.

The formal definition is as follows (after [7]): The p-th persistent homology groups of filtered complex $\mathcal{K}$ are the images of the homomorphisms induced by inclusion, $H^{i,j}(\mathcal{K}) = imf^{i,j}$, where $f^{i,j} : \mathcal{K}_i \hookrightarrow \mathcal{K}_j$.

We want to remind a theorem saying when persistence of two filtered complexes are equal:

Theorem 4.1 (Persistence equivalence theorem, [7])**.** *Consider persistent homology of two filtered complexes X and Y. Let $\phi_i : H_*(X_i) \to H_*(Y_i)$:*

$$\begin{array}{ccccccccc} \dots H_*(X_0) & \longrightarrow & H_*(X_1) & \longrightarrow & \dots & \longrightarrow & H_*(X_{n-1}) & \longrightarrow & H_*(X_n)\dots \\ \downarrow{\scriptstyle \phi_0} & & \downarrow{\scriptstyle \phi_1} & & & & \downarrow{\scriptstyle \phi_{n-1}} & & \downarrow{\scriptstyle \phi_n} \\ \dots H_*(Y_0) & \longrightarrow & H_*(Y_1) & \longrightarrow & \dots & \longrightarrow & H_*(Y_{n-1}) & \longrightarrow & H_*(Y_n)\dots \end{array}$$

If the ϕ_i are isomorphisms and all the squares commute, then the persistent homology of X and Y is the same.

5. Discrete Morse Theory.

5.1. **Morse matching and Morse graph.** In this section a brief introduction to discrete Morse theory is given. For further theoretical details please consult [10, 17]. Let us have a complex $\mathcal{K}$. Discrete Morse theory partitions the cells of $\mathcal{K}$ into *matched cells* and *critical cells*. The critical cells, together with a boundary operator we describe later, form a chain complex called the *Morse complex* of $\mathcal{K}$. Importantly, this Morse complex has homology isomorphic with the homology of the initial complex.

The matching is an injective partial map $M : \mathcal{K} \nrightarrow \mathcal{K}$. Each cell of $\mathcal{K}$ can be matched with exactly one of its co-faces. Some cells can remain unmatched, and are called *critical*. These cells constitute the resulting Morse complex.

Let us introduce a graph representation of the complex K along with matching M, namely the Morse graph. This graph is denoted $G(\mathcal{K}, M)$ and is a directed graph whose vertices are formed by cells of $\mathcal{K}$. A directed edge (A, b) is added whenever b is a face of A in the complex. If b is matched with A in M, the directed edge is reversed. We use $G(\mathcal{K})$ as a shorthand for $G(\mathcal{K}, \emptyset)$.

The matching M is called *acyclic* if $G(\mathcal{K}, M)$ is a directed acyclic graph. In discrete Morse we use only acyclic matchings are considered. The paths of $G(\mathcal{K}, M)$ which connect two cells of the same dimension are called $V-$paths.

A Morse complex is called *perfect* if each cell corresponds to a homology generator of the original complex. This depends on the choice of coefficients, for example in case of a field coefficients F, we can talk about F-perfect matchings [2]. Some spaces do not admit perfect matchings, for example the Dunce hat, being contractible but non-collapsible. In this case some critical cells are necessarily spurious: they do not correspond to any homology generators. One can try to construct a best possible matching, minimizing the number of critical cells. This problem is known to be NP-complete and MAXSNP-hard [16]. As we explain later, our approach works by using a series of sub-optimal matchings.

Once an acyclic Morse matching M is obtained, we proceed with computing the Morse boundary. This procedure is described in [10]. Forman [10] proved that the resulting Morse complex has isomorphic homology to the homology of the initial complex. He also provided a formula which computes the boundary of each cell in a Morse complex. Kozlov generalized these proofs to the setting of arbitrary chain complexes [17]. Later we show that a similar construction can be also used to simplify persistence.

An procedure for computing a Morse complex is outlined in Algorithm 1. Note that for algorithmic purposes we represent a chain complex by a graph.

Algorithm 1 Compute Morse complex

Input: Complex $\mathcal{K}$
Output: Resulting Morse complex $\mathbb{M}(\mathcal{K})$
1: $M :=$ acyclic Morse matching on $G(\mathcal{K})$
2: $C :=$ list of critical cells of M
3: $bd :=$ compute Morse Boundary of $G(\mathcal{K}, M)$
4: $\mathbb{M}(\mathcal{K}) := (C, bd)$

5.2. **Discrete Morse theory for filtered complexes.** In this section we will recall the basic ideas from [21]. For a filtered chain complex $\mathcal{K}$ we say that the Morse matching M is *compatible with filtration* of $\mathcal{K}$ if for every matched $A \in \mathcal{K}$, $g(A) = g(M(A))$[1]. In other words, the matchings are made *only* between elements of the same filtration level. Consequently, directed paths cannot move upwards the filtration.

The key result in [21] is that persistence of a filtered complex $\mathcal{K}$ and a Morse complex $\mathbb{M}(\mathcal{K})$ with Morse matchings compatible with filtration are the same.

[1]By $M(A)$ we denote the element matched with A.

5.3. **Computing Morse complex as a graph algorithm.** We want to stress that the entire Morse complex construction can be computed using standard graph algorithms. The chain complex $\mathcal{K}$ (with $\mathbb{Z}_2$ coefficients) can be interpreted as a graph $G(\mathcal{K})$. The whole construction can be divided into two essential parts: finding an acyclic Morse matching and computing the Morse boundary.

The constructed Morse matching can be essentially arbitrary, as long as its acyclic. In the literature [16, 18, 26] greedy strategies are usually presented. In a typical situation the aim is to minimize the number of cells of the resulting complex. As mentioned earlier the problem of constructing *a* best possible matching is NP-complete and MAXSNP-hard [16]. In our approach we build, in polynomial time, a series of Morse complexes using sub-optimal matching, which results in a *perfect iterated Morse complex*. Note that we do not contradict the hardness results, because generally this is *not* a perfect Morse complex of the initial complex.

A strategy used in [16, 18, 26] reduces the complex in BFS order. The running time is linear in the size of the graph of the input complex. This graph is initially sparse in practice. In the worst case it can grow quadratically during the course of the iterations, if all cells become connected. Therefore the cost of performing a Morse matching is $O(n^2)$, where n is the number of cells in the initial complex.

Finding an *acyclic* Morse matching is equivalent to the following problem – which directed edges in the Morse graph can be reversed so that the graph remains acyclic. This is closely related to the *minimum feedback arc set* problem. While this problem is NP-hard, efficient approximation schemes exist [9].

We believe that the main strength of our approach is the possibility of using approximate or randomized algorithms to obtain an optimal solution. We are currently investigating the possibility of adapting existing algorithms for related graph problems. We hope to report on that in the final version of this paper.

5.3.1. *Computing Morse boundaries.* Intuition: Since a face of a critical cell can be non-critical, we have to replace it by a corresponding critical cell (or cells). This correspondence is computed by traversing the Morse graph.

Let us state the problem of computing Morse boundary in terms of graph theory. The Morse boundary of a critical cell is formed by the set of critical cells which are reachable by an *odd* number of paths in the Morse graph. This is a special case of Forman's formula [10] in case of $\mathbb{Z}_2$ coefficients. Since the number of such paths can grow exponentially in the number of cells, brute force calculation is ineffective (as noted in [21]). However, this problem can be solved efficiently, exploiting the fact that the Morse graph is acyclic. One of the possibilities is to compute the boundary of each cell separately, using memoized recursion. This leads to a simple to implement algorithm, which works in pessimistic time $O(c(|V| + |E|))$, where V and E are respectively the vertices and edges of graph $G(\mathcal{K}, M)$ and c is the number of critical cells. For more efficient algorithms see [18, 12].

6. Iterated Morse Complex construction for persistent homology

In [21] it is shown that when a Morse complex is constructed based on a Morse matching compatible with filtration, the resulting Morse complex and the initial complex have isomorphic persistent homology. Here we provide an extension of this result.

Later in this paper we assume that the Morse complex construction is a functor $\mathbb{M}$ from category of filtered chain complexes to itself. It can be seen as an algorithm which computes a Morse complex of an input chain complex. We say that $\mathbb{M}$ is *vital* if it makes at least one Morse matching whenever there some possible Morse matchings exist.

Let us take a vital functor $\mathbb{M}$. We assume that the Morse matching used to construct $\mathbb{M}(\mathcal{K})$ is compatible with filtration of $\mathcal{K}$. Filtration values of the cells of $\mathbb{M}(\mathcal{K})$ are inherited from the filtration of cells in $\mathcal{K}$. We construct the *perfect iterated Morse complex*, $\mathbb{M}^\infty(\mathcal{K})$ which is the fixed point of the sequence $\mathbb{M}(\mathcal{K})$, $\mathbb{M}^1(\mathcal{K})$, $\mathbb{M}^2(\mathcal{K}), \ldots$, obtained by iterating functor $\mathbb{M}$.

Algorithmically, a single iteration of our simplification procedure computes of Morse complex. In case of filtered complexes, the performed Morse matchings have to be compatible with filtration.

Algorithm 2 describes the entire simplification procedure. See Figure 1 for a detailed example illustrating this algorithm. To compute persistence based on simplified complex $\mathbb{M}^\infty(\mathcal{K})$ use the matrix reduction algorithm.

Algorithm 2 Simplification for persistence computations

Input: Initial filtered complex $\mathcal{K}$
Output: Simplified complex $M^\infty(\mathcal{K})$, having the same persistent homology
1: $\mathbb{M}^0(\mathcal{K}) = \mathcal{K}$
2: **repeat**
3: $\quad \mathbb{M}^{i+1}(\mathcal{K}) :=$ Morse complex of $\mathbb{M}^i(\mathcal{K})$ using matchings compatible with filtration
4: **until** $\mathbb{M}^{i+1}(\mathcal{K}) = \mathbb{M}^i(\mathcal{K})$ (which means it is a perfect iterated Morse complex)
5: **return** $\mathbb{M}^{i+1}(\mathcal{K})$

If n is the size of the initial complex, at most n iterations are needed to obtain a perfect iterated Morse complex. Each iteration takes time $O(n^2)$, which is the worst-case behavior of the Morse matching and Morse boundary algorithms. Therefore the overall worst-case complexity is cubic. This is a very rough estimate. Based on the experiments in [18] we expect to have a reasonable simplification for the first iteration, therefore consecutive iterations would operate on a much smaller complex. This is to be verified numerically.

Let us remind that obtaining a perfect Morse complex is impossible in some cases and computationally hard in general. On the other hand, in the *iterated* Morse decomposition setting, we always achieve the desired perfect iterated Morse complex.

We show that each cell in $\mathbb{M}^\infty(\mathcal{K})$ either creates or kills a feature of *non-zero* persistence. Since we are interested in reducing the number of cells, the resulting complex is the minimal complex encoding persistence of the original complex.

In [21] it is only assumed that the filtered chain complex is input. Since the resulting Morse complex is itself a (filtered) chain complex, one can iterate this construction. The main strength of our approach is based on the following novel observation.

Lemma 6.1. If $\mathbb{M}$ is vital and the Morse matchings are compatible with filtration, then the resulting complex $\mathbb{M}^\infty(\mathcal{K})$ has the following property: For every $A \in \mathbb{M}^\infty(\mathcal{K})$ and for every $b_1, \ldots, b_k$ in boundary of A we have $g(A) > g(b_1), \ldots, g(b_k)$.

Proof. We know that $g(b_i) \leq g(A)$, as we use sublevel-set filtration. Now, if there existed b_i in the boundary of A such that $g(b_i) = g(A)$, then a Morse matching would have been made between A and b_i[2]. This would contradict the vitality assumption of $\mathbb{M}$. □

Having this property of the resulting complex $\mathbb{M}^\infty(\mathcal{K})$, we can now present the main theorem of this paper:

Theorem 6.1. *Let $\mathbb{M}^\infty(\mathcal{K})$ be the perfect iterated Morse complex obtained from the initial filtered chain complex $\mathcal{K}$ by iterative construction of Morse complexes using Morse matchings compatible with filtration. Then:*
$\mathcal{K}$ and $\mathbb{M}^\infty(\mathcal{K})$ have the same persistence. (Correctness)
Every cell $A \in \mathbb{M}^\infty(\mathcal{K})$ either starts or terminates a non-zero length *persistence interval.* (Optimality)

Proof. Correctness: To show that the algorithm is correct we will apply the Persistence Equivalence Theorem for each iteration:

[2]Note that this is true because we use coefficients in a field.

$$\begin{array}{ccccccc} \ldots & \xrightarrow{j_{l-1}} & H_*(\mathbb{M}_l^i(\mathcal{K})) & \xrightarrow{j_l} & H_*(\mathbb{M}_{l+1}^i(\mathcal{K})) & \xrightarrow{j_{l+1}} & \ldots \\ \downarrow & & \downarrow bd_l^{\mathbb{M}} & & \downarrow bd_{l+1}^{\mathbb{M}} & & \downarrow \\ \ldots & \xrightarrow{k_{l-1}} & H_*(\mathbb{M}_l^{i+1}(\mathcal{K})) & \xrightarrow{k_l} & H_*(\mathbb{M}_{l+1}^{i+1}(\mathcal{K})) & \xrightarrow{k_{l+1}} & \ldots \end{array}$$

We need to show two things:

(1) That the vertical maps are isomorphisms on homology level.
(2) That all squares commute.

First note that the vertical maps send each chain in the input complex to a corresponding chain of the Morse complex. We remind that to find a corresponding Morse chain we follow appropriate V-paths in the Morse graph.

1) Vertical arrows are isomorphisms: this is a consequence of Theorem 11.24 from [17], which states that the upper chain complex M is decomposed by the Morse construction into an *acyclic* part and the Morse complex having homology isomorphic with M. It is necessary to use Kozlov's theory, since Forman's proofs work only for CW-complexes.

2) To prove that the squares commute for each i, l, let us take a chain $c \in \mathbb{M}_l^i = \sum c_j$. We show that $(bd_{l+1}^{\mathbb{M}} \circ j_l)(c) = (k_l \circ bd_l^{\mathbb{M}})(c)$.

Down and right $(k_l \circ bd_l^{\mathbb{M}})$: If c_j is critical, it is unchanged by the vertical map. Otherwise, we follow the paths of the Morse graph to compute the corresponding chain in the Morse complex. Repeating this computation for every c_j, the value of $bd_l^{\mathbb{M}}$ on c is obtained. Moving right with inclusion, the chain remains the same.

Right and down $(bd_{l+1}^{\mathbb{M}} \circ j_l)$: first the chain c is inserted by inclusion into level $l+1$ of filtration, so it is unchanged. But now we move with the vertical arrow, which might be richer on this level, as additional paths enter the Morse graph. Note that since we force the paths to be non-increasing with filtration, $bd_{l+1}^{\mathbb{M}}$ restricted to level l is the same as $bd_l^{\mathbb{M}}$. In other words, any V-path starting at c_j at level at most l can only reach cells of lower or equal filtration values. In particular, it will never reach any critical cell introduced at level $l+1$.

Therefore the two images of chain c are the same and the diagram commutes. Applying Persistence Equivalence Theorem finishes the proof that persistent homology is unchanged during the *iterated* Morse complex construction.

Optimality:

For the proof we consider the boundary matrix of $\mathbb{M}^\infty(\mathcal{K})$, sorted by filtration values [7]. Note that in general any column either creates or kills a feature, possibly of persistence zero [7]. Let A be a column of this matrix. From Lemma 6.1 we know that $g(A) > g(b_i)$, where b_i is the *lowest one* in column A. Suppose that column A was now reduced. The reduced column is either zero or non-zero. If it is zero, a homological feature is created. If this feature is not killed later, its persistence is infinite. If the reduced column is non-zero then is has the lowest one, b. During the reductions, the lowest ones can only move upwards in the matrix [7], so we have $g(b) \leq g(b_i) < g(A)$. The homological feature created by b is killed by A, yielding a persistent interval, $[g(b), g(A)]$, of non-zero length. This proves the optimality. □

Remark 6.2. Intuitively, each level-set (viewed as a chain complex) is optimally simplified in terms of standard homology. In particular, each cell of a level-set of $\mathbb{M}^\infty(\mathcal{K})$ corresponds to a homological feature of the corresponding level-set of $\mathbb{K}$. This follows from Lemma 6.1. Additionally, the boundary relations between level-sets are preserved.

Remark 6.3. Since each cell in $\mathbb{M}^\infty(\mathcal{K})$ encodes the information about either birth or death of exactly one feature of non-zero persistence, its removal essentially changes persistent homology. Therefore the number of cells in $\mathbb{M}^\infty(\mathcal{K})$ is minimal. Let p be the number of finite and k the number of infinite persistence intervals of $\mathcal{K}$. Then $|\mathbb{M}^\infty(\mathcal{K})| = 2p + k$. If $2p + k$ is small, computing persistence in time $O((2p + k)^3)$ can be feasible.

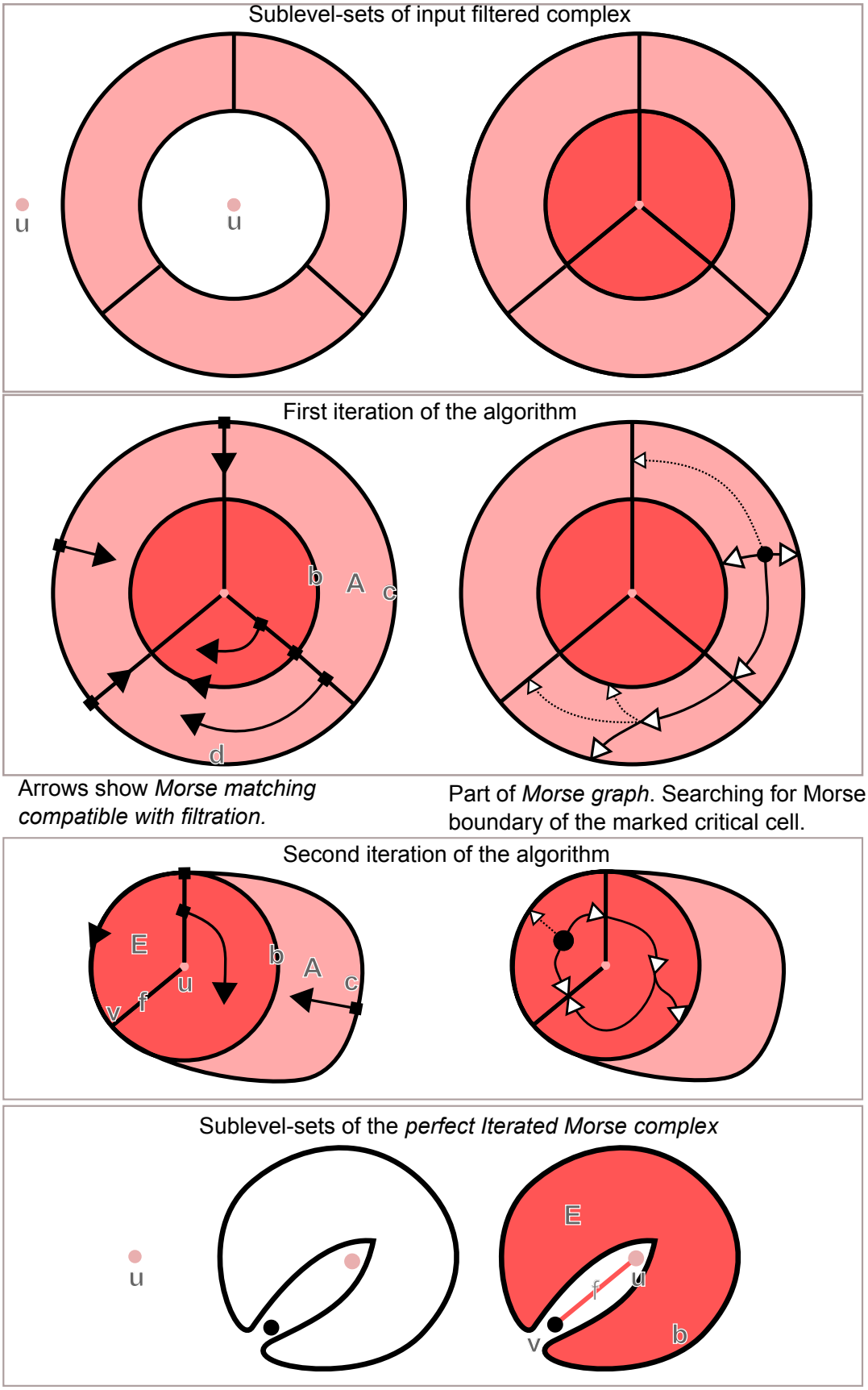

FIGURE 1. A visual representation of two iterations of the iterated Morse complex technique.

6.1. **Example.** Figure 1 illustrates various details of the presented method. The exposition is slightly informal to convey intuition to non-specialists.

The top row shows the input filtered chain complex with three filtration levels. Corresponding sublevel-sets are depicted, ordered by filtration level. When needed vertices are shown as dots, colored by filtration value, but normally we assume that a vertex appears if two or more edges intersect in the picture.

At the first level a single vertex (u) enters, creating a connected component. Next an annulus enters, adding another connected component and a 1-cycle. Finally at level three the 1-cycle is filled and the two components are merged. Since persistent homology is always preserved by our method, the resulting filtration (depicted in the final row) behaves in the same way.

Second and third row, on the left: Each arrow points from a cell to its co-face and indicates that these were matched. In accordance with the definition of the Morse matching, no two arrows start or end at the same cell.

Note that all the performed matchings are compatible with filtration, i.e. both matched cells live on the same level. At the end no more such matchings can be made, which means that the resulting complex is a perfect iterated Morse complex. In particular edge f cannot be paired with vertex v or u as they live on different levels.

Admittedly, the matchings were chosen so that the result is easy to visualize. Of course, this simple example could be optimally simplified in one iteration. An example reacquiring more iterations, e.g. containing the Dunce hat, would be harder to visualize.

We want to point out that an optimal homological simplification is performed on each level (in the sense homology of chain complexes). If we look at each level-set (not: sublevel-set) of the resulting complex, the homology is unchanged, moreover each cell corresponds to a homology class of the initial level-set. At level one the single vertex is unchanged. At level two, we have a component (reduced to a point) 1-cycle (reduced to a single edge with its boundary removed). At level two the situation is less intuitive: the only nontrivial homology is in dimension 2. After simplification an open 2-cell is left, so the homologies agree.

Second and third row, on the right: part of the Morse graph is drawn. We depict the process of computing the Morse boundary of a given critical cell. It involves following the arcs to find all reachable critical cells of dimension lower by one. The found cells which are reachable by an odd number of paths form the Morse boundary. Some of the paths, marked with dotted-arcs, end in non-critical cells, which are not interesting. The Morse boundary of cell A is the set of edges $\{b, c, d\}$. Note that the Morse boundary of cell E contains only edge b and not edge f, since there are two paths in the graph from E to f. For the same reason, vertex v is not in the Morse boundary of b (this part of the graph is not depicted).

The bottom row depicts the sublevel-sets of the resulting filtration, namely: the perfect iterated Morse complex. It is clearly visible that the resulting complex is as simple as possible, while the behavior in terms persistent homology is preserved.

7. Conclusions

In our opinion the presented preprocessing technique has several advantages:

(1) It is provably optimal, rather than a heuristic, generalizing in [24] to arbitrary dimensions. If the output is small, computing persistence will be fast.
(2) It is based on graph theory – we can use efficient algorithms (exact and approximate) and their existing implementations – in particular libraries for distributed graph operations [19].
(3) It is intuitive and easy to visualize.
(4) Since the Morse matchings never go between the filtration levels, they can be made independently in every filtration level. (No additional cycles will be introduced as the V-paths never go upward the filtration.) This can be exploited in a parallel implementation.
(5) It can be used to formalize and generalize homology-preserving properties of graph pyramids used in image recognition [22]. Moreover, they can be beneficial in verified

homology computation [15] by avoiding matrix operations which are costly to verify automatically.

We are aware of some drawbacks of our method:

(1) There may exist bad cases, where the complexity will be unsatisfactory. In particular if there are no zero-length persistence intervals, the preprocessing is futile. (This is rare in practice, where lower-star filtration is usually employed.)
(2) For some cell complexes a chosen algorithm for Morse matchings may make only a small number of matchings in each iteration which results in cubical complexity of the procedure.
(3) This technique might not be suitable for cubical data. Existing methods [12, 25] rely on the compact representation of cubical grids. In our case the complex need not be a cubical complex after the first iteration, preventing us from storing it efficiently.

Acknowledgments

The authors would like to thank Herbert Edelsbrunner and Urlich Bauer for their valuable comments and suggestions. Both authors are supported by Google Research Awards program. We thank Marian Mrozek for the supervision of this project. PD. is partially supported by grant IP 2010 046370. HW. is partially supported by Foundation for Polish Science IPP Programme "Geometry and Topology in Physical Models".

8. Appendix

8.1. **Interpretation of Morse matchings in algebraic algorithm.** In this short subsection we show how the Morse matchings can be interpreted in the language of standard algebraic algorithm [7]. We want to point out however, that the main strength of the method presented in this paper is the fact, that it can be implemented with using just graph algorithms. The intuition is given just for a clarity purposes. In the Figure 2 a simple Morse matching is interpreted in the matrix.

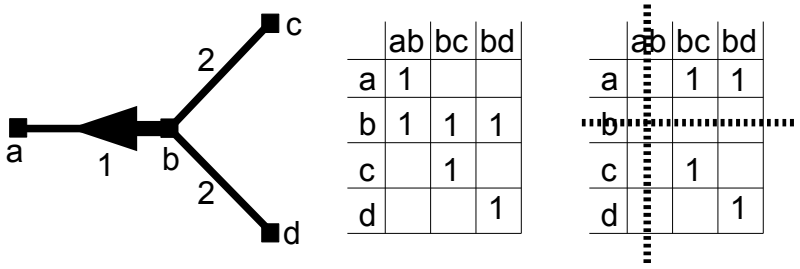

Figure 2. Left: the initial complex with a single Morse matching. Filtration value of all vertices is 1 and of the edges are given in the picture. Middle: the initial boundary matrix. Right: the boundary matrix after the matching ab with b is made. As one can see, performing the Morse matchings resolves all the possible future conflicts caused by b at once.

References

[1] A. Rathod, personal communication.
[2] R. Ayala, D. Fernndez-Ternero, J. A. Vilches, *Perfect discrete Morse functions on 2-complexes*, Pattern Recognition Letters - PRL, DOI: 10.1016/j.patrec.2011.08.011.

[3] U. Bauer, C. Lange, M. Wardetzky, *Optimal topological simplification of discrete functions on surfaces*, Discrete & Computational Geometry 47:2 (2012), 347377.
[4] C. Chen, M. Kerber, *An output-sensitive algorithm for persistent homology*, 27th Annual Symposium on Computational Geometry (SoCG 2011).
[5] C. Chen, M. Kerber, *Persistent homology computation with a twist*, 27th European Workshop on Computational Geometry (EuroCG 2011), 2011.
[6] C. J. A. Delfinado, H. Edelsbrunner, *An incremental algorithm for Betti numbers of simplicial complexes*, Proceeding SCG '93 Proceedings of the ninth annual symposium on Computational geometry. pp. 232-239.
[7] H. Edelsbrunner, J. Harer, *Computational Topology. An Introduction.* Amer. Math. Soc., Providence, Rhode Island, 2010.
[8] H. Edelsbrunner, D. Letscher and A. Zomorodian. Topological persistence and simplification. Discrete Comput. Geom. 28 (2002), 511-533.
[9] G. Even, J. Noar, B. Schieber and M. Sudan, *Approximating minimum feedback sets and multicuts in directed graphs*, Algorithmica, 20, (1998) 151-174.
[10] R. Forman, *Morse theory for cell complexes*, Advances in Mathematics, 134:90–145, 1998.
[11] P. Frosini, *A distance for similarity classes of submanifolds of a Euclidean space*, Bulletin of the Australian Mathematical Society, 42, 3 (1990), 407-416.
[12] D. Günther, J. Reininghaus, H. Wagner, I. Hotz, *Effcient Computation of 3D Morse-Smale Complexes and Persistent Homology using Discrete Morse Theory*, The Visual Computer 28(10), 959-969 (2012).
[13] A. Gyulassy, V. Natarajan, V. Pascucci, B. Hamann, *Efficient Computation of Morse-Smale Complexes for Three-dimensional Scalar Functions*, IEEE Transactions on Visualization and Computer Graphics, Vol. 13 , Issue: 6, pp. 1440 - 1447, 2007.
[14] S. Harker, K. Mischaikow, M. Mrozek, V. Nanda, *Discrete Morse Theoretic Algorithms for Computing Homology of Complexes and Maps*, submitted to Foundations of Computational Mathematics.
[15] J. Heras, M. Dénès, G. Mata, A. Mörtberg, M. Poza, V. Siles, *Towards a certified computation of homology groups for digital images*, Lecture Notes in Computer Science Volume 7309, 2012, pp 49-57.
[16] M. Joswig , M. E. Pfetsch , *Computing optimal Morse matchings*, SIAM Journal on Discrete Mathematics, Vol. 20 Issue 1, pp. 11 – 25, 2006.
[17] D. Kozlov, *Combinatorial Algebraic Topology*, Springer-Verlag, 2007.
[18] T. Lewiner, H. Lopes, G. Tavares, *Optimal discrete Morse functions for 2-manifolds*, Computational Geometry: Theory and Applications 26(3): pp. 221-233.
[19] G. Malewicz, M. Austern, A. Bik, J. Dehnert, I. Horn, N. Leiser, G. Czajkowski, *Pregel: a system for large-scale graph processing*, Proceedings of the 2010 ACM SIGMOD International Conference on Management of data, pp. 135-146, 2010.
[20] N. Milosavljevic, D. Morozov, P. Skraba, *Zigzag Persistent Homology in Matrix Multiplication Time*, Proceedings of the 27th Annual Symposium on Computational Geometry (SCG'11).
[21] K. Mischaikow, V. Nanda, *Morse theory for filtrations and efficient computation of persistent homology*, Discrete & Computational Geometry submitted.
[22] S. Peltier, A. Ion, Y. Haxhimusa, W. G. Kropatsch, G. Damiand, *Computing Homology Group Generators of Images Using Irregular Graph Pyramids*, GbRPR 2007: 283-294.
[23] V. Robins, *Towards computing homology from finite approximations*, Topology Proceedings, 24:503-532, (1999).
[24] 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, (2010) accepted.
[25] H. Wagner, C. Chen, E. Vuçini, *Efficient computation of persistent homology for cubical data*, Proceedings of the 4th Workshop on Topology-based Methods in Data Analysis and Visualization (TopoInVis 2011).
[26] H. Wagner, P. Dłotko, M. Mrozek, *Computational topology in text mining*, in M. Ferri et al. (Eds.): CTIC 2012, LNCS 7309, pp. 117127, 2012.
[27] A. Verri, C. Uras, P. Frosini, M. Ferri, *On the use of size functions for shape analysis*, Biological Cybernetics, 70, (1993), 99-107.

Institute of Computer Science, Jagiellonian University, Krakow, Poland
E-mail address: pawel.dlotko@ii.uj.edu.pl

Institute of Computer Science, Jagiellonian University, Krakow, Poland
E-mail address: hubert.wagner@ii.uj.edu.pl