Available online at www.sciencedirect.com
ScienceDirect
Computational Geometry 41 (2008) 126–148
Computational Geometry
Theory and Applications
www.elsevier.com/locate/comgeo

Localized homology ☆

Afra Zomorodian [a,*,1], Gunnar Carlsson [b]

[a] *Department of Computer Science, Dartmouth College, Hanover, NH, USA*
[b] *Department of Mathematics, Stanford University, Stanford, CA, USA*

Received 3 August 2006; received in revised form 10 October 2007; accepted 28 February 2008

Available online 13 March 2008

Communicated by K. Mehlhorn

Abstract

In this paper, we provide the theoretical foundation and an effective algorithm for localizing topological attributes such as tunnels and voids. Unlike previous work that focused on 2-manifolds with restricted geometry, our theory is general and localizes arbitrary-dimensional attributes in arbitrary spaces. We implement our algorithm to validate our approach in practice.

Keywords: Localization; Homology; Generators; Persistent homology; Mayer–Vietoris

1. Introduction

Topology describes how a space is connected, reflecting the presence of certain *qualitative* features in the space, such as the tunnel in Fig. 1(a). While topology, and specifically the formalism of *homology*, is capable of detecting the *existence* of such features, it cannot directly tell us about their *location*. It is possible to augment the classical reduction scheme [25] for homology to produce descriptions of the topological attributes. This is not done in practice, however, as the resulting descriptions are geometrically unpredictable. For example, homology may produce the long solid loop in the figure as a description of the tunnel. While topologically correct, this description is geometrically useless. The *localization problem* is determining the location of topological features within a space.

In this paper, we formulate a theoretical foundation for localization. Our theory has a computational nature, yielding an effective implementable algorithm. All previous work address the localization of one type of attribute within one type of space: tunnels within closed surfaces. In contrast, our theory is general and localizes arbitrary-dimensional attributes in arbitrary spaces.

The localization problem arises naturally in many disciplines that analyze low-dimensional geometry. Often, the topology of a space may have significant repercussions on the ability of geometric algorithms to perform effectively

☆ Research of second author partially supported by DARPA under grant HR0011-06-1-0038, and of both authors by DARPA under grant HR0011-05-1-0007.

* Corresponding author.
E-mail address: afra@cs.dartmouth.edu (A. Zomorodian).

[1] Portion done during author's post-doctoral studies at Stanford University.

0925-7721/$ – see front matter
doi:10.1016/j.comgeo.2008.02.003

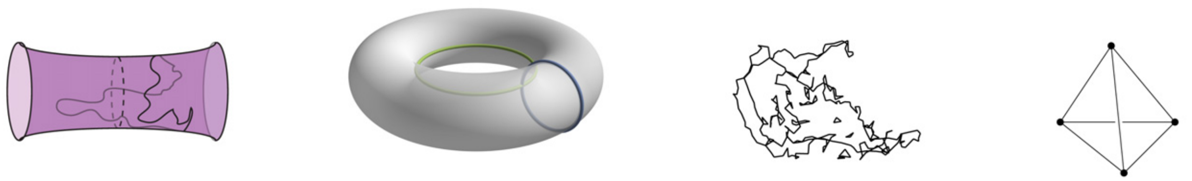


Fig. 1. (a) The localization problem. Both the solid and dashed loops describe the tunnel. But topology cannot distinguish between them. (b) Homological description. A surface of a donut, a *torus*, has $\beta_0 = 1$ component, $\beta_1 = 2$ tunnels (one outside, one inside), and encloses $\beta_2 = 1$ void. The drawn cycles are representatives for the two classes of 1-cycles. (c) A homology basis element may have multiple components. We computed this particular element in a complex representing the HIV Protease molecule. (d) A tetrahedron has four faces, but its 1-skeleton has cycles that form a vector space of dimension $\beta_1 = 3$.

or even terminate. In *computer graphics*, undersampling and noise result in extraneous topology, such as spurious handles in reconstructed surfaces [22]. For example, the Stanford head model for Michelangelo's *David* has 340 small tunnels, none of which are present in the sculpture [17]. This topological noise hinders subsequent geometry processing, such as simplification, smoothing, and parameterization. We need to locate, measure, and remove these tunnels to facilitate our geometric algorithms [17,29]. In *sensor networks*, non-uniform distribution, terrain features, or catastrophic failure of nodes may lead to regions without working sensors. These holes break efficient but greedy communication algorithms. We need to locate these holes so we may route messages around them [14].

The localization problem also appears within abstract higher-dimensional spaces. In *robotics*, a compact representation of the *configuration space* of a robot that captures the connectivity is useful for fast computation of ensemble properties, such as the probability of folding (p-fold) of a protein conformation [1]. In *shape description*, locating the topological attributes of the *tangent complex* allow us to find the features, such as corner points or edges, within the shape itself [6]. In *computer vision*, understanding the local structure of natural images can impact the design of novel compression algorithms. Recently, analysis of the nine-dimensional Mumford dataset uncovered topological structure that was previously unknown and unexploited in compression [5]. Again, we need to localize the topological attributes to discover their geometric implications.

1.1. Prior work

Researchers in computer graphics and computational geometry have mostly analyzed a limited form of the localization problem in the past, focusing on finding tunnels within closed surfaces. In the terminology of topology, they address the specific problem of describing *1-cycles* on *oriented 2-manifolds without boundary*, often with additional geometric restrictions such as embeddedness, smoothness, or having a Riemannian structure. Based on different notions of optimality, there have been various approaches to this problem, such as growing regions on the surface [17], searching within the associated *Reeb graph* [29], cutting a surface into a disk either along the *canonical polygonal schema* [21,28] or the *cut-graph* [12], finding a *system of loops* corresponding to a minimal presentation of the *fundamental group* of the surface [7], or obtaining a shortest set of loops that generate the fundamental group or the first homology group [13].

The prior results exploit two structural assumptions. They search for a one-dimensional attribute or a *loop*, generally represented with a set of edges; and they search within a two-dimensional closed surface, generally represented with a triangulation, where each edge is in exactly two triangles. These are simplifying assumptions that allow for tailored algorithms. Some of these algorithms may be extended for more general spaces [11] such as surfaces with non-manifold structure or portions of different dimensions, such as the space in Fig. 12. However, our result is the first that generalizes to localizing higher-dimensional attributes, such as enclosed volumes, and searching within higher-dimensional space, such as localizing the voids in the solid in Fig. 13.

Recently, Freedman and Chao take the approach of measuring attributes by the minimum enclosing ball in the ambient space [15]. They show that this notion of size allows for a greedy algorithm to find an optimal basis. For attributes of co-dimension greater than one—such as loops in three dimensions—this is a weak measure, so the optimal result may not provide tight localized descriptions.

1.2. Our work

We address the localization problem in the most general setting. Our space is simply any *topological space*, a set of points where each point knows its neighbors. Also, we attempt to localize arbitrary-dimensional attributes. We place no restrictions on the geometry, such as requiring a Euclidean metric, but include it via a *cover*: a set of spaces that contain the original space in their union. The key advantage of this approach is the decoupling of geometry from topology. We allow the domain specialist to design and experiment with different covers on different spaces to fulfill the requirements of a particular application.

Given a cover, our task in this paper is to localize the topological attributes *relative to* that cover. We emphasize, therefore, that *locality* is defined with respect to the cover, and different covers give different results. Our focus in this paper is on extracting the geometric information contained in a cover. We show, however, that even simple covers give information and may be used recursively to construct more geometric covers that yield tighter descriptions.

Algebraic topology can be described roughly as the study of spaces through their algebraic images [18]. A fundamental tool for this study is a *functor*, a map that not only forms algebraic images of the spaces themselves, but also of maps between them. Functors play a key role in *exact sequences*, machinery that allow deduction of properties about spaces. These sequences are not algorithmic in nature and they must be worked out by hand on a case by case basis. We believe the key contribution of our paper is the insight that the recent theory of persistent homology [9,31] is in fact an algorithmic view of functoriality. This paper is an application of this insight, but it has a much broader scope and may be applied to convert other functors into powerful tools for computation using a computer.

In this paper, we introduce *localized homology* as a synthesis of persistent homology with the classical blowup construction [27]. Our theory is general and works for all covers and spaces. More importantly, our theory has a computational nature, so we derive a simple algorithm and validate it through experiments. While our method is grounded in theory, the main ideas are accessible, so we begin with a complete non-technical overview in the next section. We then provide the technical details to formalize our method and derive the algorithm. As part of this process, we show that the persistence algorithm works for a broad class of cell complexes and we give a specification that computes descriptions of non-boundary cycles. We end the paper with some experimental results where our algorithm localizes topology in spaces where other algorithms fail.

2. Overview

In this section, we motivate our method by examining the nature of the localization problem, introducing a possible solution, exposing the difficulties through simple examples, and resolving them through an algebraic approach. We begin with an intuitive description of homology.

2.1. Homology

Homology is the topological invariant that is frequently used in practice as it is computable by linear algebraic methods in all dimensions. This method characterizes the connectivity of a space X through the structure of its holes [18], studying them via equivalence classes of *cycles* called *homology classes*. A *homology cycle* has an intuitive meaning in $\mathbb{R}^3$: A 0-*cycle* is a formal linear combination $\sum_{x\in X} r_x x$ of points in the space and can be interpreted as an oriented multiset of points x with multiplicity $|r_x|$ and orientation $\mathrm{sign}(r_x)$. A 1-*cycle* is a loop or union of loops going around a tunnel. And a 2-*cycle* is a surface that encloses an empty space. We obtain homology classes from cycles by passing to a quotient vector space H_k by a subspace of *boundaries*. For instance, a 1-dimensional homology class is an equivalence class of 1-cycles, where two cycles are equivalent provided there is a two-dimensional surface in X whose boundary is the union of the two cycles, viewed as a union of loops. The *kth Betti number* β_k of the space is the dimension of the homology vector space in dimension k. The 1-cycles in Fig. 1(b) represent two classes which form a basis for H_1 of the *torus*, the surface of a donut.

Any vector space has many equivalent bases. To compute a homology basis, we transform boundary matrices using the *standard reduction scheme*, an extension of *Gaussian elimination* to coefficient sets other than $\mathbb{R}$ [25]. Indeed, all known algorithms for computing homology are a variant of this scheme, including the *persistence algorithm* that we utilize in this paper. This reduction is inherently non-local as we have no geometric control over the selection: the chosen basis is often geometrically ugly and may have multiple components, such as the basis element shown in

(a) Graph (b) (c) (d)

Fig. 2. A graph (a) and three possible bases (b, c, d) for its H_1. Only (b) is local.

Fig. 1(c). Since homology computes a basis without regard to geometry, any cycle in the 1-skeleton of a tetrahedron in Fig. 1(d) is a candidate basis element and not those that are geometrically "local". Similarly, for the graph in Fig. 2(a), homology may choose one of the non-local bases (c, d) instead of the local one (b). The examples demonstrate that homology does not favor localization by nature. It has no knowledge of the geometry of the space, so it cannot identify local bases.

2.2. Measures and covers

We wish to provide geometric knowledge to homology for selecting local bases. To do so, we must first define what we mean by the word *local*. Intuitively, the idea is to determine which homology classes arise from small parts of a space and which are more global. There are many ways to formulate this notion. Suppose that our space is equipped with a metric. One approach is by defining a norm on the homology classes and defining small classes to be local given that norm. For instance, we could measure the *diameter* of the image of the classes within the space, such as the length of the shortest loop for 1-cycles [13] or the area of the smallest enclosing void for 2-cycles. Alternatively, we could define the radius of the minimum enclosing ball to be the measure of a cycle [15]. In this paper, we develop a flexible framework that incorporates many different methods. Let us begin with a preliminary definition of what we mean by local.

Definition 1. Let X be a topological space and $\mathcal{U} = \{X^i\}_i$ be a collection of subsets $X^i \subseteq X$ such that $X \subseteq \bigcup_i X^i$. A homology class is $\mathcal{U}$*-local* iff it is in the image of the induced homomorphism $H_k(X^i) \to H_k(X)$, where H_k is the kth homology group.

As we shall see, our final definition is slightly different than this definition, although we may derive descriptions corresponding to this definition easily. The collection $\mathcal{U}$ *covers* the space and its pieces provide us with a notion of what locality means: a homology class of the space is local if it exists in one of the pieces of the cover. This notion of size is well-adapted to the minimum enclosing ball measure above. We could let $\mathcal{U}$ be the set of all balls of radius $\leqslant r$, in which case the norm would simply be the smallest value of r such that the class is $\mathcal{U}$-local.

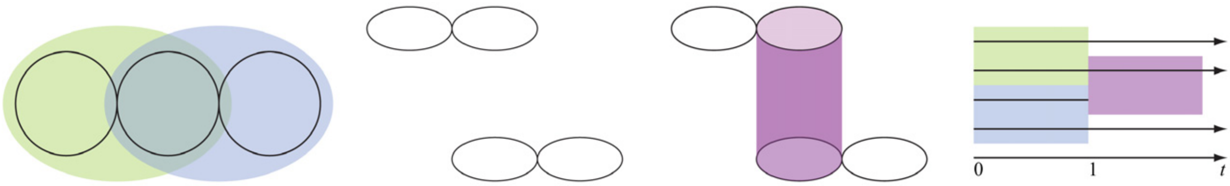
Given a covering $\mathcal{U}$, the goal of this paper is to point out that persistent homology can be used to determine whether a homology class is $\mathcal{U}$-local. We also obtain an algorithm for generating a homology basis that is localized with respect to the cover. We do not directly address the problem of choosing useful covers. This is because the notion of size may be application-dependent, in which case we may wish to use different methods for generating covers. We illustrate with a few examples.

Example 1 *(hypercubes).* Suppose our space X is embedded in $\mathbb{R}^n$. We may then consider a covering $X_{N,r}$ of $\mathbb{R}^n$ that consists of overlapping hypercubes of length $1/N + 2r$, centered at vertices of a lattice of size $1/N$. To get a cover, we intersect the hypercubes with X.

Example 2 *(landmarks).* If our space X is metric, we may sample a finite set of *landmark* points $L \subseteq X$ and define a Voronoi cell $V(p)$ for $p \in L$ by

$$V(p) = \{x \in X \mid \mathrm{d}(x, p) \leqslant \mathrm{d}(x, p'), \forall p' \in L\}.$$

The collection $\mathcal{U} = \{V(p)\}_{p \in L}$ is a cover. Note that this construction only requires a metric on the space itself and not in the ambient space, so it may be used for non-Euclidean cases, such as when the metric is tree-based.



(a) Space and cover (b) Local pieces ($t = 0$) (c) Blowup ($t = 1$) (d) Persistence barcode

Fig. 3. Our approach. Given a space equipped with a cover (a), we first blow up the space into local pieces (b) and then glue back the pieces to get the blowup complex (c), giving us a filtration consisting of two complexes at times $t = 0$ and $t = 1$, respectively. The persistence barcode (d) localizes the topology of the original space with respect to the cover.

Example 3 *(Morse functions).* Let X be the torus, embedded in $\mathbb{R}^3$ on its side with $h : X \to \mathbb{R}$ its z coordinate. We may now cover the torus with overlapping slices, $h^{-1}([k/N - r, (k + 1)/N + r])$ for fixed values of N and r. Note that classes that are local with respect to this cover are local only in the direction of h.

In summary, our framework is flexible enough to incorporate different notions of size through the notion of covers. In the future, we hope to address specific methods of generating covers for different applications. In this paper, we focus on computing a localization, given a cover.

2.3. Our framework

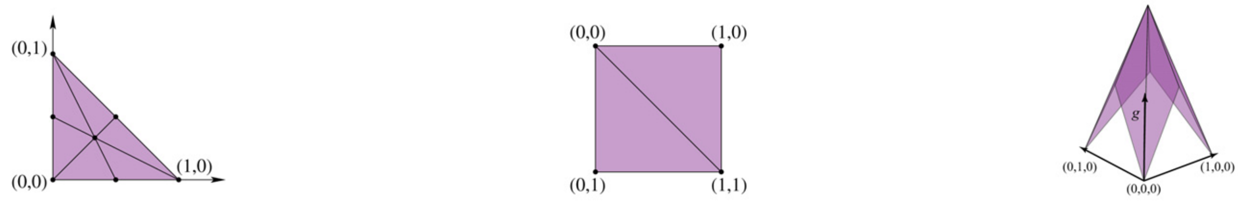
We begin by applying by blowing up the space into local pieces according to the cover. For example, the graph containing three cycles covered by two sets in Fig. 3(a) is blown up into two pieces in (b), each with two 1-cycles. Since the middle cycle of the original space is contained in the intersection of the cover sets, it exists in both local pieces. To recover the global topology, we equate the two copies of the middle cycle by gluing a cylinder to them. The resulting construction in Fig. 3(c), which we call the *Mayer–Vietoris blowup complex*, has the same number of cycles as the original space but also incorporates the geometric cover information within its structure.

We now need to compute homology bases for the blowup complex that are compatible with bases for the local pieces. Fortunately, the theory of *persistent homology* furnishes the required bases [31]. We incrementally assemble the blowup complex so that the local pieces are included at time 0 and the cylinder is sewn in at time 1, completing the structure. Persistence computes compatible homology bases across this growth history. Therefore, it can track individual basis elements, representing their lifetimes in a multiset of intervals called a barcode [6]. The barcode for our example, shown in Fig. 3(d), has three half-infinite intervals, corresponding to the three 1-cycles in both the original space and its blowup complex. But we can also color the barcode to show where the 1-cycles are located. There are four intervals at time < 1, representing the four local 1-cycles in Fig. 3(b). At time 1, the cylinder equates the two copies of the middle 1-cycle, so one of the two intervals that represent the two copies ends. The choice of the interval corresponds to the choice of the basis representative of the middle cycle lying in either of the two sets of the cover. As the two are homologous, the choice is arbitrary, but of course, we may choose the one that is geometrically more pleasant.

To summarize, given a space equipped with a cover, we incorporate the geometry contained within the cover into homology by building the blowup complex and computing its persistent homology. We call this method *localized homology*. Our localization naturally reflects the quality of the given cover, and covers that reflect the geometry of the space give better descriptions. We will show, however, that even naive covers give quick and useful information about the location of attributes.

3. Background

In this section, we review the algebraic tools required in our work. As it is infeasible to include a complete treatment, we sketch some of the basic ideas and include formal constructions only when necessary. For a more complete account, we refer to standard texts in the area [16,18] as well as the cited papers. We organize this section as a flow of ideas for capturing the topology of the spaces in Section 3.1: from global (3.2), to local (3.3), to product (3.4), to persistent (3.5).



(a) $|K^2| = \Delta^2$ (b) Product of two edges (c) $g(\Delta^2)$

Fig. 4. (a) The shaded standard 2-simplex Δ^2 is the convex hull of the labeled vertices. Subdividing Δ^2 using the barycenters of the edges and the triangle gives the simplicial complex K^2 whose underlying space is Δ^2. (b) The product of two edges [0, 1] is the shaded quadrangle. Using $<$ as the order on the vertices gives the product simplicial complex shown. (c) The function $g : \Delta^2 \to \mathbb{R}$ is defined in Section 4.1 using the subdivision complex K^2 in (a): g is 0 on the vertices of Δ^2, 1 on the edge barycenters, 2 on the triangle barycenter, and is extended linearly elsewhere.

3.1. Topological space

The fundamental object in topology is a *topological space*, an abstraction of a metric space. Rather than using a metric to define *open sets*, a topological space X comes equipped with a set of open sets that define its connectivity. A subset $X_0 \subseteq X$ can be given the subspace topology whose open sets are the intersections of X_0 with the open sets in X. The, we call X_0 a *subspace*, and (X, X_0) a *pair*. A family $\mathcal{U} = \{X^i\}_i$ of subspaces $X^i \subseteq X$ is a *cover* (*covering*) of X if $X \subseteq \bigcup_i X^i$. We say $\mathcal{U}$ covers X.

Suppose we have topological spaces X and Y and continuous maps $f : X \to Y$ and $g : Y \to X$ between them. If gf and fg are equal to the identity maps on the respective spaces, the spaces are *homeomorphic* and have the same *topological type*: $X \approx Y$. This is the most restrictive notion of equivalence in topology. We get a relaxation through the notion of *homotopy*. Given two maps $f_0, f_1 : X \to Y$, if there is a continuous map $h : X \times [0, 1] \to Y$ such that $h(x, 0) = f_0(x)$ and $h(x, 1) = f_1(x)$, then f_0 and f_1 are *homotopic* via *homotopy* h; $f_0 \simeq f_1$. Now, for our maps f and g above, if gf and fg are merely homotopic to the respective identities, then X and Y are *homotopy equivalent*: $X \simeq Y$.

For computation, we need a combinatorial structure for representing a topological space. Let $[n] = \{0, 1, \ldots, n\}$ be the first $n + 1$ natural numbers. A *n-simplex* σ is the convex hull of $n + 1$ affinely independent *vertices* $S = \{v^i\}_{i \in [n]}$ in $\mathbb{R}^d$, $d \geqslant n$. A simplex τ defined by $T \subseteq S$ is a *face* of σ. A *simplicial complex* K is a finite set of simplices that meet along faces, all of which are in K. A *subcomplex* of K is a subset $L \subseteq K$ that is also a simplicial complex. The *k-skeleton* $K^{(l)}$ of K is the subcomplex containing simplices with dimension less than or equal to l. The *underlying space* $|K|$ of a simplicial complex K is $|K| = \bigcup_{\sigma \in K} \sigma$. A *triangulation* of a topological space X is a simplicial complex K such that $|K| \approx X$.

There is a standard realization for an n-simplex as follows. Let e^0 be the origin in $\mathbb{R}^n$ and $e^i = (0, \ldots, 1, \ldots, 0)$, $1 \leqslant i \leqslant n$, be the ith *standard basis vector* for $\mathbb{R}^n$ with a 1 in the ith position and 0's elsewhere. The *standard n-simplex* Δ^n is the convex hull of $\{e^i\}_{i \in [n]}$. The shaded triangle in Fig. 4(a) is the standard 2-simplex. For any *indexing set* $J \subseteq [n]$, Δ^J is the face of Δ^n spanned by $\{e^j\}_{j \in J}$. Note that $\Delta^{[n]} = \Delta^n$. The standard simplex may be subdivided using the barycenters of its faces to produce the simplicial complex K^n with $|K^n| = \Delta^n$. Each non-empty face Δ^J of Δ^n has an associated vertex v^J in K^n. Δ^J is triangulated by subcomplex $K^J \subseteq K^n$ with $|K^J| = \Delta^J$. Fig. 4(a) displays this subdivision on the standard 2-simplex.

3.2. Homology

For a topological space X, the *homology groups* $H_n(X)$ are a family of Abelian groups for integers $n \geqslant 0$ with the following properties [18]:

Functoriality: Each H_n is a *functor*, that is, for any continuous map $f : X \to Y$, there is an induced homomorphism $H_n(f) : H_n(X) \to H_n(Y)$, such that $H_n(fg) = H_n(f)H_n(g)$ and $H_n(i_X) = i_{H_n(X)}$, where i is the identity.

Homotopy invariance: If $f, g : X \to Y$ are homotopic, then $H_n(f) = H_n(g)$. If f is a homotopy equivalence, then $H_n(f)$ is an isomorphism.

For any field F, there is a version of homology with coefficients in F that takes values in F-vector spaces. Throughout this paper, we always compute over field coefficients. The dimension of the vector space is the *nth Betti number* $\beta_n(X)$ of the space.

3.3. Subspace homology

Since we are interested in localizing homology, we need to understand the relationship between local and global homology of a space. The algebraic gadget that elucidates this relationship is the *Mayer–Vietoris sequence*. To define it, we first need to discuss the notion of an *exact sequence*.

Definition 2 *(long exact sequence).* A sequence

$$A \xrightarrow{f} B \xrightarrow{g} C$$

of homomorphisms of Abelian groups (or linear transformations of vector spaces over a field F) is said to be *exact* if $gf \equiv 0$ and the kernel of g is equal to the image of f as subgroups (or subspaces) of B. A sequence of homomorphisms (or linear transformations)

$$\cdots \to A_{i+1} \xrightarrow{f_{i+1}} A_i \xrightarrow{f_i} A_{i-1} \xrightarrow{f_{i-1}} A_{i-2} \xrightarrow{f_{i-2}} \cdots$$

is exact if each subsequence $A_{k+1} \to A_k \to A_{k-1}$ is exact. Such a sequence is called a *long exact sequence*.

As an aside, we note that the second requirement of an exact sequence clearly implies the first, but it is customary within algebraic topology to itemize requirements according to increasing restrictive power, a practice we follow in our formalism. A long exact sequence morally means that if one has information about two of every three consecutive groups in the sequence, one can obtain information about the third. One such sequence, the Mayer–Vietoris, relates the homology in the intersection of subspaces to that of the individual pieces and that of the whole space.

Theorem 1 *(Mayer–Vietoris). Suppose we have a topological space X with cover $\{Y, Z\}$ such that the interiors of Y and Z also cover X. Then, there exists a long exact sequence*

$$\cdots \longrightarrow H_i(Y \cap Z) \longrightarrow H_i(Y) \oplus H_i(Z) \longrightarrow H_i(X) \longrightarrow H_{i-1}(Y \cap Z) \longrightarrow H_{i-1}(Y) \oplus H_{i-1}(Z) \longrightarrow \cdots.$$

3.4. Product homology

The Mayer–Vietoris sequence allows for computation by hand, but is not algorithmic. Fortunately, there is a geometric counter-part, the *Mayer–Vietoris blowup* that is defined as a subspace of a product space. For topological spaces X and Y, the *product space* $X \times Y$ is also a topological space. Given simplicial complexes X and Y, we triangulate the product space $|X \times Y|$ with a simplicial complex $X \times Y$ as follows [19, page 262]. Assume we are given total orderings $<_X$ and $<_Y$ on the vertex sets V_X and V_Y, respectively. The vertex set of $X \times Y$ then is $V_{X\times Y} = V_X \times V_Y$. And a subset $\sigma \subseteq V_{X\times Y}$ spans a simplex iff there exist simplices $\sigma_X \in X$ and $\sigma_Y \in Y$ so that $\sigma \subseteq \sigma_X \times \sigma_Y$ and the restriction of the partial order $<_X \times <_Y$ to σ is a total ordering. It is easy to construct a homomorphism from $|X \times Y|$ to $|X| \times |Y|$.

Example 4 *(two edges).* Fig. 4(b) displays the product of two edges $X = [0, 1]$ and $Y = [0, 1]$ with $<$ as the total order on each vertex set. The product simplicial complex has $V_{X\times Y} = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$ and an edge from $(0, 0)$ to $(1, 1)$ since $(0, 0) < (1, 1)$ but $(1, 0) \nless (0, 1)$ and $(0, 1) \nless (1, 0)$.

The connectivity of a product space is clearly related to the connectivity of its factors, as made explicit by the following theorem. Below, C_* is the *chain complex*:

$$\cdots \to C_{k+1} \xrightarrow{\partial_{k+1}} C_k \xrightarrow{\partial_k} C_{k-1} \to \cdots,$$

where C_k is the free Abelian group on oriented simplices in the complex, and $\partial_k : C_k \to C_{k-1}$ is the boundary homomorphism.

Theorem 2 *(Alexander–Whitney). Let X and Y be simplicial complexes with vertex orderings $<_X$ and $<_Y$ defining the product simplicial complex $X \times Y$. For maps of simplicial complexes that preserve the vertex orderings, there are natural chain maps*

$$C_*(X \times Y) \underset{\mathcal{S}}{\overset{\mathcal{A}}{\rightleftarrows}} C_*(X) \otimes C_*(Y), \tag{1}$$

that induce isomorphisms on homology groups. Here, $\mathcal{A}$ is the Alexander–Whitney *map and $\mathcal{S}$ is the* shuffle *homomorphism* [4]. *For pairs (X, X_0) and (Y, Y_0), we get a relative version*

$$C_*\big(X \times Y, (X \times Y_0) \cup (X_0 \times Y)\big) \underset{\mathcal{S}}{\overset{\mathcal{A}}{\rightleftarrows}} C_*(X, X_0) \otimes C_*(Y, Y_0). \tag{2}$$

In other words, we can obtain the homology of a product space from the homology of its factors through the Alexander–Whitney map.

3.5. Persistent homology

As sketched in Section 2.3, we put together the blowup complex incrementally to see the local and global topologies at different times. For a topological space X, this construction gives a *filtration* $\{X^n\}_{n \geqslant 0}$, a nested sequence of subspaces

$$\emptyset = X^0 \subseteq X^1 \ldots \subseteq X.$$

We then call X a *filtered space*. We may similarly filter a simplicial complex to obtain a *filtered complex*. Over fields, each space X^j has a kth homology group $H_k(X^j)$, a vector space whose dimension $\beta_k(X^j)$ counts the number of topological attributes in dimension k. Viewing a filtration as a growing space, we see that topological attributes appear and disappear. If we could track an attribute through the filtration, we could talk about its *lifetime* within this growth history. The theory of *persistent homology* validates this intuition [31]. A filtration yields a *directed space*

$$\emptyset = X^0 \overset{i}{\hookrightarrow} \cdots \overset{i}{\hookrightarrow} X^j \cdots \overset{i}{\hookrightarrow} X,$$

where the maps i are the respective inclusions. Applying the kth dimensional homology functor H_k from Section 3.2 to both the spaces and the maps, we get another directed space

$$\emptyset = H_k\big(X^0\big) \xrightarrow{H_k(i)} \cdots \xrightarrow{H_k(i)} H_k\big(X^j\big) \xrightarrow{H_k(i)} \cdots \xrightarrow{H_k(i)} H_k(X)$$

where $H_k(i)$ are the respective induced maps. Persistent homology states that any directed vector space has a simple description [31]. For an interval $[a, b]$, $a \in \mathbb{Z}^+$, $b \in \mathbb{Z}^+ \cup \{\infty\}$, let $F[a, b] = \{F_i\}_{i \geqslant 0}$ be a directed vector space over field F that is equivalent to F within the interval and empty elsewhere, and with identity transformations $F_i \to F_j$ within the interval. Under suitable finiteness hypotheses that are satisfied for all our spaces, the homology directed space may be written as a direct sum

$$\bigoplus_{i=0}^{s} F[a_i, b_i],$$

where the description is unique up to reordering of summands. In layman's terms, persistent homology states we may indeed track topological attributes and measure their lifetimes as intervals $[a_i, b_i]$. The *persistence barcode* is the finite multiset of lifetime intervals [6]. We can compute barcodes for arbitrary-dimensional simplicial spaces over arbitrary fields using the persistence algorithm [31].

We also need to compute descriptions of the homology cycles. The original paper on persistence did not discuss descriptions, although the algorithm could compute descriptions for cycles that eventually became boundaries [9] and these descriptions were used subsequently for computing linking numbers [10]. Our subsequent theoretical extension also did not discuss descriptions [31]. For this paper, we show that the persistence algorithm works for a broad class of cell complexes and we enable it to compute descriptions of non-boundary cycles. We describe this extension in Section 4.4.

4. Localized homology

In this section, we formalize our approach. To be mathematically rigorous, we begin in Section 4.1 by giving a definition for localized homology in the most general setting: topological spaces and singular homology. We do this to place our work within the traditional framework of algebraic topology. This definition, however, is not immediately useful for computation as singular homology deals with infinite-dimensional spaces. So, in Section 4.2, we give a combinatorial definition for simplicial complexes. While it is known that singular and simplicial homology are equivalent, the same is not known of our two definitions, so we next show the simplicial definition gives the same results as the singular definition. In Section 4.3, we construct a chain complex that yields equivalent barcodes and once again show its equivalence to the previous definition. This last definition allows us to specify a natural basis and the boundary operator for the chain complex in Section 4.5. We may now construct a filtration directly from our space and cover. Feeding this filtration into the persistence algorithm, we get our localized solution. We note for the specialist that an equivalent route entails giving definitions for simplicial sets and cellular homology.

4.1. Singular definition

Given an arbitrary topological space equipped with a cover, we *blow up* the space to incorporate the geometry contained in the cover: Each piece of the space expands according to the number of cover sets it falls within.

Definition 3 *(Mayer–Vietoris blowup complex).* Given a topological space X with a cover $\mathcal{U} = \{X^i\}_{i\in[n-1]}$ of $n = \operatorname{card} \mathcal{U}$ sets, let $X^J = \bigcap_{j\in J} X^j$ for $J \subseteq [n-1]$. The *blowup complex* $X^{\mathcal{U}} \subseteq X \times \Delta^{n-1}$ of X and $\mathcal{U}$ is

$$X^{\mathcal{U}} = \bigcup_{\emptyset\neq J\subseteq[n-1]} X^J \times \Delta^J. \tag{3}$$

$X^{\mathcal{U}}$ is equipped with two natural projection maps $\pi_X : X^{\mathcal{U}} \to X$ and $\pi_\Delta : X^{\mathcal{U}} \to \Delta^{n-1}$ given by the inclusion $X^{\mathcal{U}} \hookrightarrow X \times \Delta^{n-1}$ followed by projection onto the respective factors [27, page 108].

While the construction is standard, there is no standard terminology, so we have adopted our own for which we provide justification at the end of this section.

Example 5 *(cover with two sets).* Suppose X comes with cover $\mathcal{U} = \{X^0, X^1\}$ as shown in Fig. 5(a), where we represent X as an interval and draw ellipses to indicate the extent of the cover sets. The cover defines the intersection piece $X^{\{0,1\}} = X^{[1]}$. The blowup $X^{\mathcal{U}}$ is a subset of $X \times \Delta^1$ as shown in Fig. 5(b), where we draw Δ^1 as the interval [0, 1]. Following Eq. (3), $X^{\mathcal{U}}$ is the union of three pieces, corresponding to the three local regions the cover defines:

$$\begin{aligned}
&X^0 \times \Delta^{\{0\}} = X^0 \times \{0\},\\
&X^1 \times \Delta^{\{1\}} = X^1 \times \{1\},\\
&X^{[1]} \times \Delta^{[1]} = X^{[1]} \times [0, 1].
\end{aligned}$$

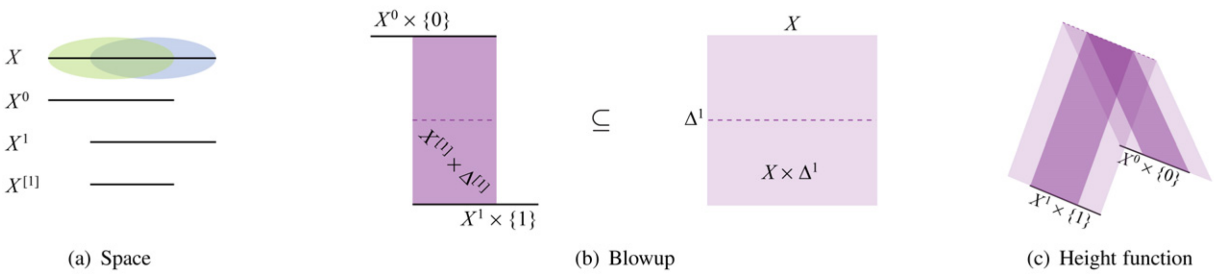


Fig. 5. Blowup complex. (a) The cover $\mathcal{U} = \{X^0, X^1\}$ for space X also defines intersection $X^{\{0,1\}} = X^{[1]}$. (b) The blowup $X^{\mathcal{U}} \subseteq X \times \Delta^1$ is the union of three pieces shown. Here, Δ^1 is visualized as interval [0, 1]. (c) The function f on $X^{\mathcal{U}} \subseteq X \times \Delta^1$.

Example 6 *(cover with three sets).* Suppose our space is a single point and our cover is three sets that contain it. Then, the blowup is a triangle, as shown in Fig. 4(a). We use this simple example extensively, especially in proving one of the main theorems in the paper.

In constructing the blowup complex, we simply stretch certain pieces, so we don't tear or glue. Clearly then, the blowup complex has the same topology as the original space. We formalize this next.

Lemma 1 *(global). The projection $\pi_X : X^{\mathcal{U}} \to X$ is a homotopy equivalence in the following cases*:

- *$\mathcal{U}$ is an open covering of a normal space, e.g. any subspace of $\mathbb{R}^n$,*
- *$\mathcal{U}$ is a covering of a simplicial complex by subcomplexes.*

Therefore, π_X induces an isomorphism at the homology level. That is, $X^{\mathcal{U}} \simeq X$ and $H_*(X^{\mathcal{U}}) \cong H_*(X)$.

Proof. The proof essentially follows from Segal, whose result is for the case of coverings admitting a subordinate partition of unity [27, Theorem 4.1]. This occurs when the covering is *numerable*. It is quite standard that these hypotheses hold for open coverings of spaces homeomorphic to finite CW-complexes. Moreover, one can easily show that they also hold for coverings of finite simplicial complexes by subcomplexes. □

Throughout this paper, we assume our cover satisfies the requirements of Lemma 1. We now define a function f on $X^{\mathcal{U}}$ that assembles the pieces such that the persistent homology of the resulting filtration gives the localization. We first define a function g on Δ^{n-1} by utilizing its triangulation K^{n-1}, assigning a value to the centroid of each face and interpolating linearly.

Definition 4 *(height functions f, g).* For the face $\emptyset \neq \Delta^J$ of Δ^{n-1}, let v^J be the associated vertex in K^{n-1}. Define $g : \Delta^{n-1} \to \mathbb{R}$ linearly on the complex with $g(v^J) = \dim \Delta^J$, and on Δ^{n-1} by identification. Define $f : X^{\mathcal{U}} \to \mathbb{R}$ by the composition $X^{\mathcal{U}} \xrightarrow{\pi_\Delta} \Delta^{n-1} \xrightarrow{g} \mathbb{R}$.

Fig. 4(c) illustrates g on Δ^2 and we see f on the blowup complex in Fig. 5(c). We filter the blowup complex using f.

Definition 5 *(filtered blowup).* Let $X_t^{\mathcal{U}} = f^{-1}([0, t])$. The *filtered blowup complex* is the family $\{X_t^{\mathcal{U}}\}_{t \geqslant 0}$.

In other words, when we visualize f as a height function on $X^{\mathcal{U}}$ as in the figures, $X_t^{\mathcal{U}}$ is everything in $X^{\mathcal{U}}$ below height t.

Example 7 *(cover with two sets).* In Example 5, we constructed the blowup complex for a space X with a two-set cover $\mathcal{U} = \{X^0, X^1\}$, as shown in Fig. 5. Filtering the complex, we have:

$$X_0^{\mathcal{U}} = X^0 \times \{0\} \dot{\cup} X^1 \times \{1\},$$
$$X_1^{\mathcal{U}} = X_0^{\mathcal{U}} \cup X^{[1]} \times [0, 1]/\sim,$$

where $\sim$ is the identification along $X^{[1]} \times \{0, 1\}$ shown in Fig. 5(c).

Example 8 *(cover with three sets).* In Example 6, X is a point, $\mathcal{U}$ consists of three sets that contain X, and $X^{\mathcal{U}}$ is K^2. Using function g in Fig. 4(c), we may filter the blowup to get the filtered blowup complex $X_t^{\mathcal{U}}$, shown for the specified values in Fig. 6. Note that the complex changes topologically only at integer values for t. This should be intuitively clear from the construction.

At time 0, the blowup complex contains the local pieces of X. For Example 5, $X_0^{\mathcal{U}}$ consists of the two segments shown at the bottom of the hat shape in Fig. 5. For Example 8, $X_0^{\mathcal{U}}$ consists of the three copies of X shown in Fig. 6(a). We formalize this concept next.

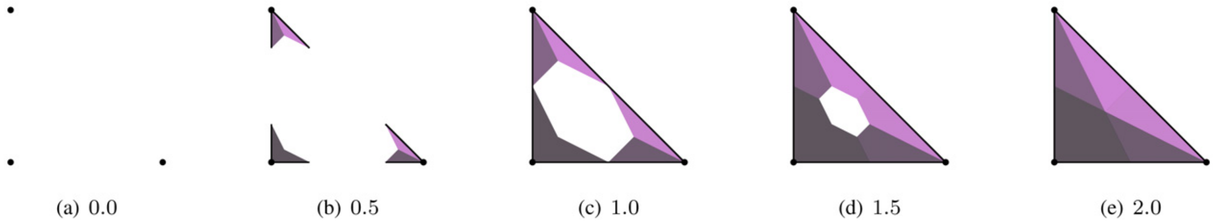


Fig. 6. Singular blowup $X_t^{\mathcal{U}}$ for specified t in Example 8. The space X is a point and its cover $\mathcal{U}$ is three sets that contain it.

Lemma 2 *(local). The space $X_0^{\mathcal{U}}$ is the disjoint union of the local pieces of the space, i.e. $X_0^{\mathcal{U}} \approx \dot{\bigcup}_{i\in[n-1]} X^i$. Therefore,*

$$H_k\big(X_0^{\mathcal{U}}\big) \cong \bigoplus_{i\in[n-1]} H_k\big(X^i\big),$$

that is, we get the homology of the local pieces at time 0.

Proof. The corresponding chain complexes clearly split as the stated direct sum, and homology preserves direct sums. □

So, we capture the local homology at time 0. At time $n-1$, the incremental construction is complete and $X_{n-1}^{\mathcal{U}} = X^{\mathcal{U}}$. Therefore, Lemma 1 asserts that $X_{n-1}^{\mathcal{U}}$ has the global homology of X. We may now state our definition for localization.

Definition 6 *(localized homology).* Given a topological space X and a cover $\mathcal{U} = \{X^i\}_{i\in[n-1]}$, let $i : X_0^{\mathcal{U}} \hookrightarrow X_{n-1}^{\mathcal{U}}$ be inclusion, inducing $\iota_* : H_*(X_0^{\mathcal{U}}) \to H_*(X_{n-1}^{\mathcal{U}})$. The *$\mathcal{U}$-localized homology of X* is the image of ι_*. We say that a homology class in the image is *$\mathcal{U}$-local*.

In contrast to our initial definition (Definition 1), this definition allows a homology class to be a sum of cycles in different pieces in the cover, but is also algorithmic. Applying persistent homology to the filtration, we get barcodes that describe the relationship between the local and global homology of the space. The $\mathcal{U}$-local homology classes are those that exist at time 0 and continue to exist till time $n-1$. These classes correspond to persistence barcode intervals that contain both 0 and $n-1$.

We note that our definition for localized homology only uses the inclusion $X_0^{\mathcal{U}} \hookrightarrow X_{n-1}^{\mathcal{U}}$ and not the intermediate filtration levels $X_t^{\mathcal{U}}$. We now show that the intermediate filtrations also carry useful information. We begin by creating coarser versions of a given cover.

Definition 7 *($\mathcal{U}[l]$).* Given a topological space X, a cover $\mathcal{U} = \{X^i\}_{i\in[n-1]}$, and $0 < l \leqslant n$, we define $\mathcal{U}[l]$ to be the cover of X whose elements are the $\binom{n}{l}$ l-fold unions of elements of $\mathcal{U}$. That is, $\mathcal{U}[l] = \{\mathcal{U}^J\}_J$, where $J \subseteq [n-1]$ with card $J = l$ and $\mathcal{U}^J = \bigcup_{j\in J} X^j$.

The following result relates these coarser covers to intermediate filtrations of the blowup complex.

Theorem 3 *(intermediate). A $\mathcal{U}[l]$-local class in $H_k(X)$ lies in the image of the homomorphism*

$$H_k\big(X_{l-1}^{\mathcal{U}}\big) \to H_k\big(X_{n-1}^{\mathcal{U}}\big) \cong H_k(X). \tag{4}$$

Proof. By Definition 6, a k-dimensional class is $\mathcal{U}[l]$-local iff it lies in the image of the homomorphism

$$H_k\big(X_0^{\mathcal{U}[l]}\big) \to H_k\big(X_{m-1}^{\mathcal{U}[l]}\big), \tag{5}$$

where $m = \binom{n}{l}$. By Lemma 2, the domain of homomorphism (5) is

$$H_k\big(X_0^{\mathcal{U}[l]}\big) \cong \bigoplus_{\substack{J\subseteq[n-1]\\ \text{card } J = l}} H_k\big(\mathcal{U}^J\big),$$

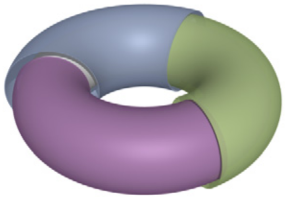


Fig. 7. A torus covered with three sets that overlap pairwise.

where $\mathcal{U}^J = \bigcup_{j\in J} X^j$ is a set in the coarsened cover. By definition and by Lemma 1, the co-domain of homomorphism (5) is

$$H_k\big(X_{m-1}^{\mathcal{U}[l]}\big) = H_k\big(X^{\mathcal{U}[l]}\big) \cong H_k(X).$$

Therefore, we may rewrite homomorphism (5) as

$$\bigoplus_{\substack{J\subseteq[n-1]\\ \operatorname{card} J=l}} H_k\big(\mathcal{U}^J\big) \to H_k(X).$$

That is, a class is $\mathcal{U}[l]$-local if it is a sum of classes h_j which lie in homomorphisms $H_k(\mathcal{U}^J) \to H_k(X)$, respectively. Since the image of homomorphism (4) is a subgroup, it suffices to show that each class h_j is contained in it. Let $W = \mathcal{U}^J$ with covering $\mathcal{W} = \{X^j\}_{j\in J}$. There is an evident inclusion $j : W^{\mathcal{W}} \hookrightarrow X^{\mathcal{U}}$ so that the following diagram commutes:

$$\begin{tikzcd} H_k(W^{\mathcal{W}}) \arrow[r, "H_k(j)"] \arrow[d, "H_k(\pi_W)"'] & H_k(X^{\mathcal{U}}) \arrow[d, "H_k(\pi_X)"] \\ H_k(W) \arrow[r, "\varphi"] & H_k(X) \end{tikzcd}$$

The homomorphism $H_k(\pi_W)$ is an isomorphism by Lemma 1, so it follows immediately that h_j is in the image of the homomorphism φ iff it is in the image of the homomorphism $H_k(\pi_X) \circ H_k(j)$ and we know the former is true. But the image of j is clearly contained within $H_k(X_{l-1}^{\mathcal{U}})$, which gives the result. □

The converse to the above theorem is false, as the following counter-example demonstrates.

Example 9 *(counter).* Consider the cover of the torus in Fig. 7. The sets in the cover intersect pairwise, but not triplewise, so clearly $X_1^{\mathcal{U}} \approx X_2^{\mathcal{U}} \simeq X$. Therefore, the generator in $H_2(X)$ is in the image of $H_2(X_1^{\mathcal{U}})$, but is not $\mathcal{U}[2]$-local as it cannot come from the homology of any twofold union of the sets in the cover: each set is a cylinder with the homotopy type of a circle and vanishing H_2. In algebraic topological terms, the filtration gives an *obstruction* to a class being $\mathcal{U}[l]$-local, but for $l \geqslant 2$, it is not a complete obstruction.

We end this section by giving an explicit relationship between the blowup complex and the Mayer–Vietoris sequence, justifying the former's name. Consider the cover with two sets in Example 5. We write the Mayer–Vietoris sequence for this space by setting $Y = X^0$, $Z = X^1$, and therefore $Y \cap Z = X^{[1]}$ in Theorem 1:

$$\cdots \longrightarrow H_i\big(X^0\big) \oplus H_i\big(X^1\big) \longrightarrow H_i(X) \longrightarrow H_{i-1}\big(X^{[1]}\big) \longrightarrow \cdots.$$

Note that we only write three groups in the sequence as it has a three-fold periodicity: groups that differ in multiples of three in the sequence differ only in dimension. Alternatively, we may write a sequence using the blowup complex. The long exact sequence for the pair $(X_1^{\mathcal{U}}, X_0^{\mathcal{U}})$ computed in Example 7 is:

$$\cdots \longrightarrow H_i\big(X_0^{\mathcal{U}}\big) \longrightarrow H_i\big(X_1^{\mathcal{U}}\big) \longrightarrow H_i\big(X_1^{\mathcal{U}}, X_0^{\mathcal{U}}\big) \longrightarrow \cdots, \tag{6}$$

where $H_i(X_1^{\mathcal{U}}, X_0^{\mathcal{U}})$ is the reduced homology of the quotient space $X_1^{\mathcal{U}}/X_0^{\mathcal{U}}$ [18, page 115]. We now relate two sequences. By Lemma 2, the first terms match. The second terms also match by Lemma 1 as $X_1^{\mathcal{U}} = X^{\mathcal{U}}$. For the third term of sequence (6), we have

$$H_i\big(X_1^{\mathcal{U}}, X_0^{\mathcal{U}}\big) \cong H_i\big(X^{[1]} \times [0,1], X^{[1]} \times \{0,1\}\big),$$

by the *Excision Theorem* [18, page 119]. But by a standard application of a *Künneth Formula* [18, page 218], the right hand side is isomorphic to $H_{i-1}(X^{[1]})$, showing that the third terms are isomorphic. To reiterate, the long exact sequence for the pair composed of pieces of the blowup complex has the form of the Mayer–Vietoris sequence for the space.

4.2. Simplicial definition

The definitions in the last section assumed that X was a topological space, so the homology groups were all *singular* homology groups rather than those attached to a simplicial complex. Singular homology is defined on all topological spaces as the homology of a complex of infinite dimensional vector spaces over the ground field. Therefore, it is not directly computable from the definitions. On the other hand, one can prove numerous properties of the construction which allows computation by hand in many cases. In order to make homology accessible to machine computation, we need to deal with finite structures. When our space is equipped with additional structure, such as being a *simplicial complex*, we can compute finite structures that agree with singular homology in the following sense: there is a canonical isomorphism from the simplicially computed homology to the singular homology. In this section, we redefine the blowup for simplicial complexes, producing a space which is itself a simplicial complex, and is therefore amenable to finite computation. We assume that we are given a simplicial complex X that represents a space of interest. We also restrict the cover $\mathcal{U}$ to consist of subcomplexes of X. Our task is twofold: we need to triangulate the blowup complex $X^{\mathcal{U}}$ and show that the simplicial homology of the resulting simplicial complex gives the same localization as the singular method.

We begin by triangulating $X^{\mathcal{U}}$. Eq. (3) states that $X^{\mathcal{U}}$ is a union of pieces of form $X^J \times \Delta^J$. Both terms $X^J = \bigcap_{j\in J} X^j$ and Δ^J are simplicial, giving us a product of simplicial complexes. We impose total orderings on the vertices of X and Δ to triangulate each product as in Section 3.4. We now define both the simplicial blowup and its filtration at once.

Definition 8 *(filtered simplicial blowup).* Let X be a simplicial complex and $\mathcal{U} = \{X^i\}_{i\in[n-1]}$ be a cover consisting of n subcomplexes. For $J \subseteq [n-1]$, let $X^J = \bigcap_{j\in J} X^j$ and

$$X_t^{\mathcal{U}} = \bigcup_{\substack{J\subseteq[n-1]\\ 0<\operatorname{card} J\leqslant t+1}} X^J \times \Delta^J. \tag{7}$$

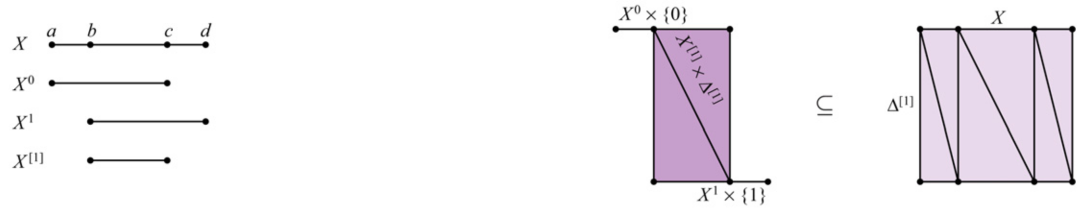
The *blowup complex of X and $\mathcal{U}$* is $X^{\mathcal{U}} = X^{\mathcal{U}}_{n-1} \subseteq X \times \Delta^{n-1}$ with projections $\pi_X : X^{\mathcal{U}} \to X$ and $\pi_\Delta : X^{\mathcal{U}} \to \Delta^{n-1}$. The *filtered blowup complex* is the family $\{X_t^{\mathcal{U}}\}_{t\geqslant 0}$.

Definition 7 mimics the singular Definitions 4 and 5. Note that $X_t^{\mathcal{U}}$ is defined for all $t \in \mathbb{R}, t \geqslant 0$, but the complex changes only at integer values.

Example 10 *(cover with two sets).* Fig. 8 constructs the blowup complex for Example 5 in simplicial form. The piece $X^{[1]} \times [0, 1]$ is completed at time 1 in Fig. 5(c), and the triangulation of the corresponding piece $X^{[1]} \times \Delta^{[1]}$ in Fig. 8(b) also arrives at time 1.

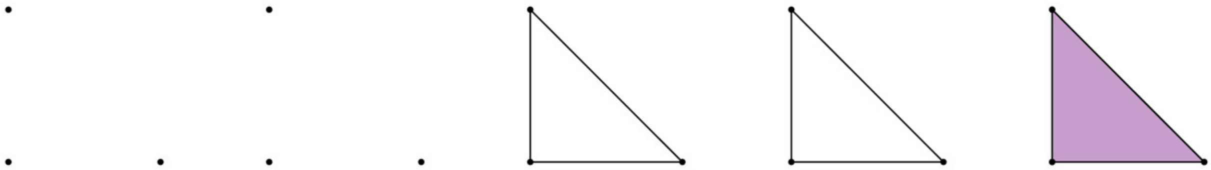
Example 11 *(cover with three sets).* Fig. 9 shows the filtered simplicial blowup complex $X_t^{\mathcal{U}}$ for the space in Examples 6 and 8. Compare with the singular blowup complex in Fig. 6. Note that the simplicial definition allows changes only at integral t values.

To complete our task, we need to show that the new simplicial definition has the same structure as the singular one from the last section. The underlying space $|X|$ of X is a topological space with the cover $|\mathcal{U}| = \{|X^i|\}_{i\in[n-1]}$. Compare the singular and simplicial filtered blowup complexes in Figs. 6 and 9. Clearly, the blowups are homotopic at all t, with topology changes at integer values only. We formalize this next.



(a) Simplicial cover (b) Blowup

Fig. 8. (a) The simplicial cover $\mathcal{U} = \{X^0, X^1\} = \{ac, bd\}$ for simplicial complex X defines $X^{\{0,1\}} = X^{[1]} = bc$. (b) The simplicial blowup complex $X^{\mathcal{U}} \subseteq X \times \Delta^1$.



(a) 0.0 (b) 0.5 (c) 1.0 (d) 1.5 (e) 2.0

Fig. 9. Simplicial blowup $X_t^{\mathcal{U}}$ for specified t in Example 11. The space X is a point and its cover $\mathcal{U}$ is three sets that contain it. Compare with the singular version in Fig. 6.

Theorem 4. *Given a simplicial complex X and a cover $\mathcal{U} = \{X^i\}_{i\in[n-1]}$ of subcomplexes, let $|\mathcal{U}| = \{|X^i|\}_{i\in[n-1]}$. There exists a canonical homeomorphism $\varphi : |X^{\mathcal{U}}| \to |X|^{|\mathcal{U}|}$ with restriction $\varphi_t : |X_t^{\mathcal{U}}| \to |X|^{|\mathcal{U}|}$ such that*:

1. *For $t \geqslant 0$, $\varphi_t(|X_t^{\mathcal{U}}|) \subseteq |X|_t^{|\mathcal{U}|}$.*
2. *For $t \in [n-1]$, φ_t is a homeomorphism onto its image.*
3. *For $t \geqslant 0$, φ_t is a homotopy equivalence.*

Proof. We begin by constructing the map φ. Note that the construction of a triangulation of a product is equipped with a canonical homeomorphism

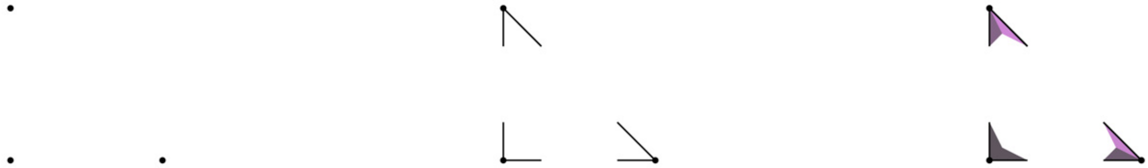
$$\alpha(X) : |X \times \Delta| \to |X| \times |\Delta|,$$

such that if we have any map $f : X \to Y$ of simplicial complexes, then we have the following commutative diagram:

$$\begin{tikzcd} |X \times \Delta| \arrow[r, "\alpha(X)"] \arrow[d, "|f\times \mathrm{id}_\Delta|"'] & |X| \times |\Delta| \arrow[d, "|f|\times|\mathrm{id}_\Delta|"] \\ |Y \times \Delta| \arrow[r, "\alpha(Y)"] & |Y| \times |\Delta| \end{tikzcd}$$

These diagrams, applied to the various inclusions $X^J \hookrightarrow X$, show that the restriction of $\alpha(X)$ to the subspace $|X^{\mathcal{U}}|$ has its image in $|X|^{|\mathcal{U}|}$. This restriction now gives the required map φ. We now examine the three statements in the theorem.

1. Given the construction above, it is clear that its restriction φ_t to $|X_t^{\mathcal{U}}|$ has its image in $|X|_t^{|\mathcal{U}|}$.
2. Recall that a continuous map between compact Hausdorff spaces that is a bijection on points is a homeomorphism [24]. Since any finite simplicial complex is a compact space, and any closed subspace of a compact space is compact, it is immediate that both $|X^{\mathcal{U}}|$ and $|X|^{|\mathcal{U}|}$ are compact Hausdorff spaces, and consequently, so are the subspaces $|X_t^{\mathcal{U}}|$ and $|X|_t^{|\mathcal{U}|}$. Therefore, it will suffice that φ_t is an injection on points for integer values of t, which is clearly the case from the definitions.
3. We begin with some useful notation, describing the portions of spaces that are allowed by g inside the singular blowup complex.



(a) $Y(0) = |X^{\mathcal{U}}_{0.5}|$ (b) $Y(1)$ (c) $Y(2) = |X|^{|\mathcal{U}|}_{0.5}$

Fig. 10. The sequence of spaces $Y(l)$ that shows the homotopy equivalence of (a) $|X^{\mathcal{U}}_t|$ to (c) $|X|^{|\mathcal{U}|}_t$ at $t = 0.5$. Note that there is a deformation retraction from the right to the left, which is the homotopy equivalence.

$$(\Delta^J)_t = \{x \in |\Delta^J|, g(x) \leqslant t\},$$
$$(\Delta^J)^{(l)}_t = \{x \in |(\Delta^J)^{(l)}|, g(x) \leqslant t\}.$$

Above, $(\Delta^J)^{(l)}$ denotes the l-skeleton of Δ^J, as defined in Section 3.1. Below, we use $\lfloor t \rfloor$ to denote the *floor of* t, the largest integer smaller than or equal to t. We may now write equivalent expressions for Definitions 5 and 7, respectively.

$$|X^{\mathcal{U}}_t| = \bigcup_J |X^J| \times (\Delta^J)^{(\lfloor t \rfloor)},$$
$$|X|^{|\mathcal{U}|}_t = \bigcup_J |X^J| \times (\Delta^J)_t.$$

There is a natural map $|X^{\mathcal{U}}_t| \to |X|^{|\mathcal{U}|}_t$ that is induced by the various inclusions on the second factors $i_J : (\Delta^J)^{(\lfloor t \rfloor)} \to (\Delta^J)_t$. Now, observe that if $\dim(\Delta^J) \leqslant \lfloor t \rfloor$, then $(\Delta^J)^{(\lfloor t \rfloor)} = (\Delta^J)_t$. For example, at $t = 1.5$, both complexes in Figs. 9(d) and 6(d) have complete 1-skeletons. If $\dim(\Delta^J) > \lfloor t \rfloor$, the barycenter of Δ^J is not contained in $(\Delta^J)_t$, so we may produce a deformation retraction ρ^J_t from $(\Delta^J)_t$ to $(\Delta^J)^{(\lfloor t \rfloor)}$ by radial projection to the boundary of Δ^J from its barycenter. In Fig. 6(d), for instance, we may retract the surface to the boundary of the triangle, the complex in Fig. 9(d). The deformation retraction and its inverse are the homotopy equivalence we need.

We now define a sequence of spaces

$$Y(l) = \bigcup_J |X^J| \times (\Delta^J)^{(\max(l, \lfloor t \rfloor))}_t,$$

as shown in Fig. 10 for our example with $t = 0.5$. Note that

$$Y(0) = |X^{\mathcal{U}}_t|,$$
$$Y(n-1) = |X|^{|\mathcal{U}|}_t,$$

so if we can show that the inclusion $Y(l) \hookrightarrow Y(l+1)$ is a homotopy equivalence for each l, we are done. To show this, we note that

$$Y(l+1) = Y(l) \cup \bigcup_{\dim \Delta^J = l+1} |X^J| \times (\Delta^J)_t.$$

For example, we add portions of the edges in going from Fig. 10(a) to Fig. 10(b). For each J, we can apply $\mathrm{id}_{X^J} \times \rho^J_t$ to the space $|X^J| \times (\Delta^J)_t$ to obtain a retraction into the subspace

$$\simeq Y(l) \cup \bigcup_{\dim \Delta^J = l+1} |X^J| \times (\Delta^J)^{(l)}{}_t$$
$$= Y(l).$$

This prove the inductive step and we are done. Completing our example, in Fig. 10, we retract the edge fragments to the vertices, and the surface pieces to the edges, in turn. □

The following corollary follows immediately from the homotopy equivalence.

Corollary 1. *The map φ of filtered spaces in Theorem* 4 *induces an isomorphism of directed Abelian groups from $H_k(|X_t^{\mathcal{U}}|)$ to $H_k(|X|_t^{|\mathcal{U}|})$ for non-negative integers k and $t \geqslant 0$. So, their barcodes are equivalent.*

Instead of using the singular definition in the last section, we may now use the simplicial definition. The former definition ensures our concepts are general, applicable to not only simplicial spaces, but other spaces as well.

4.3. Chain complex definition

Given the filtered simplicial blowup complex, persistent homology is computed using the associated chain complex. In this section, we define a smaller chain complex that gives equivalent barcodes and is computed directly from the complex and the cover.

We begin by examining the chain complex attached to the simplicial blowup complex. It follows directly from Eq. (7) that the filtered chain complex for the blowup complex is the family $\{C_*(X^{\mathcal{U}})_t\}_{t \geqslant 0}$, where $C_*(X^{\mathcal{U}})_t \subseteq C_*(X \times \Delta^{n-1})$ is

$$C_*\big(X^{\mathcal{U}}\big)_t = \sum_{\substack{J \subseteq [n-1] \\ 0 < \operatorname{card} J \leqslant t+1}} C_*\big(X^J \times \Delta^J\big), \tag{8}$$

and $C_*(X^{\mathcal{U}}) = C_*(X^{\mathcal{U}})_{n-1}$. For each piece $C_*(X^J \times \Delta^J)$, we triangulated $X^J \times \Delta^J$ in the previous section. To avoid triangulating the product, we define a smaller chain complex.

Definition 9 *(filtered blowup chain complex).* Let X be a simplicial complex and $\mathcal{U} = \{X^i\}_{i \in [n-1]}$ be a cover of n subcomplexes. For $J \subseteq [n-1]$, let $X^J = \bigcap_{j \in J} X^j$ and $C_*^{\mathcal{U}}(X)_t \subseteq C_*(X)(X) \otimes C_*(\Delta^{n-1})$ be

$$C_*^{\mathcal{U}}(X)_t = \sum_{\substack{J \subseteq [n-1] \\ 0 < \operatorname{card} J \leqslant t+1}} C_*\big(X^J\big) \otimes C_*\big(\Delta^J\big). \tag{9}$$

The *blowup chain complex of X and $\mathcal{U}$* is $C_*^{\mathcal{U}}(X) = C_*^{\mathcal{U}}(X)_{n-1}$. The *filtered blowup chain complex* is the family $\{C_*^{\mathcal{U}}(X)_t\}_{t \geqslant 0}$.

The two filtered complexes defined by Eqs. (8) and (9) give the same results. Observe that both complexes sum over the same variable J, and the summand of the first complex $C_*(X^J \times \Delta^J)$ maps naturally via the Alexander–Whitney to the summand of the second complex $C_*(X^J) \otimes C_*(\Delta^J)$, according to Eq. (1).

Definition 10. Let $\mathcal{A} : C_*(X^{\mathcal{U}}) \to C_*^{\mathcal{U}}(X)$ be the map whose the restriction to any summand $C_*(X^J \times \Delta^J)$ in Eq. (8) is the Alexander–Whitney map with values in $C_*(X^J) \otimes C_*(\Delta^J)$.

There is at most one map $\mathcal{A}$ that satisfies the requirements as we specify the map on a generating family of subcomplexes. The existence $\mathcal{A}$ is guaranteed by two facts: the naturality of the Alexander–Whitney map and $C_*(X_0 \cap X_1) = C_*(X_0) \cap C_*(X_1)$ for subcomplexes $X_0, X_1 \subseteq X$. Clearly, $\mathcal{A}(C_*(X^{\mathcal{U}})_t) \subseteq C_*^{\mathcal{U}}(X)_t$, so $\mathcal{A}$ is a chain map. We now prove that the two chain complexes give equivalent barcodes.

Theorem 5. *Given simplicial space X and simplicial cover $\mathcal{U} = \{X^i\}_{i \in [n-1]}$, there is a chain map $\mathcal{A} : C_*(X^{\mathcal{U}}) \to C_*^{\mathcal{U}}(X)$ that induces an isomorphism of directed Abelian groups from $H_k(C_*(X^{\mathcal{U}})_t)$ to $H_k(C_*^{\mathcal{U}}(X)_t)$ for non-negative integers k and $t \geqslant 0$. Consequently, the barcodes of the two chain complexes are equivalent.*

Proof. It suffices to show that the restriction $\mathcal{A}_t : C_*(X^{\mathcal{U}})_t \to C_*^{\mathcal{U}}(X)_t$ induces an isomorphism on homology groups on the integer values of t where topological changes occur. To prove this, we will perform an induction on t. Recall the notion of an *short exact sequence* of chain complexes [18]. Observe that we have a commutative diagram of chain complexes:

$$\begin{array}{ccccccccc}
0 & \longrightarrow & C_*(X^{\mathcal{U}})_{t-1} & \longrightarrow & C_*(X^{\mathcal{U}})_t & \longrightarrow & C_*(X^{\mathcal{U}})_t/C_*(X^{\mathcal{U}})_{t-1} & \longrightarrow & 0 \\
 & & \downarrow & & \downarrow & & \downarrow & & \\
0 & \longrightarrow & C_*^{\mathcal{U}}(X)_{t-1} & \longrightarrow & C_*^{\mathcal{U}}(X)_t & \longrightarrow & C_*^{\mathcal{U}}(X)_t/C_*^{\mathcal{U}}(X)_{t-1} & \longrightarrow & 0
\end{array}$$

Therefore, we have a long exact sequence:

$$\begin{array}{ccccccccc}
\cdots & \longrightarrow & H_i(C_*(X^{\mathcal{U}})_{t-1}) & \longrightarrow & H_i(C_*(X^{\mathcal{U}})_t) & \longrightarrow & H_i(C_*(X^{\mathcal{U}})_t/C_*(X^{\mathcal{U}})_{t-1}) & \longrightarrow & \cdots \\
 & & \downarrow & & \downarrow & & \downarrow & & \\
\cdots & \longrightarrow & H_i(C_*^{\mathcal{U}}(X)_{t-1}) & \longrightarrow & H_i(C_*^{\mathcal{U}}(X)_t) & \longrightarrow & H_i(C_*^{\mathcal{U}}(X)_t/C_*^{\mathcal{U}}(X)_{t-1}) & \longrightarrow & \cdots
\end{array}$$

It follows from the *Five Lemma* [18, page 129] that if the homomorphisms

$$\begin{aligned}
&H_i\big(C_*\big(X^{\mathcal{U}}\big)_{t-1}\big) \longrightarrow H_i\big(C_*^{\mathcal{U}}(X)_{t-1}\big), \\
&H_i\big(C_*\big(X^{\mathcal{U}}\big)_t/C_*\big(X^{\mathcal{U}}\big)_{t-1}\big) \longrightarrow H_i\big(C_*^{\mathcal{U}}(X)_t/C_*^{\mathcal{U}}(X)_{t-1}\big)
\end{aligned} \tag{10}$$

are isomorphisms for all i, then the homomorphisms

$$H_i\big(C_*\big(X^{\mathcal{U}}\big)_t\big) \to H_i\big(C_*^{\mathcal{U}}(X)_t\big)$$

are isomorphisms for all i. It is immediate that we can do induction on t provided we can prove that homomorphisms (10) are isomorphisms for all i and t. In turn, it suffices to prove that the following induced map on subquotients induces an isomorphism on homology groups:

$$\hat{\mathcal{A}}_t : C_*\big(X^{\mathcal{U}}\big)_t/C_*\big(X^{\mathcal{U}}\big)_{t-1} \to C_*^{\mathcal{U}}(X)_t/C_*^{\mathcal{U}}(X)_{t-1}.$$

From Eq. (8) we have a direct sum decomposition for the domain of $\hat{\mathcal{A}}_t$:

$$C_*\big(X^{\mathcal{U}}\big)_t/C_*\big(X^{\mathcal{U}}\big)_{t-1} \cong \bigoplus_{\substack{J\subseteq[n-1] \\ \operatorname{card} J=t+1}} C_*\big(X^J \times \Delta^J, X^J \times \partial\Delta^J\big), \tag{11}$$

where ∂ is the boundary operator. Similarly, from Eq. (9) we have a direct sum decomposition for the co-domain of $\hat{\mathcal{A}}_t$:

$$C_*^{\mathcal{U}}(X)_t/C_*^{\mathcal{U}}(X)_{t-1} \cong \bigoplus_{\substack{J\subseteq[n-1] \\ \operatorname{card} J=t+1}} C_*\big(X^J\big) \otimes C_*\big(\Delta^J, \partial\Delta^J\big). \tag{12}$$

The restriction of $\hat{\mathcal{A}}_t$ to each summand in Eq. (11) maps the summand to the corresponding summand in Eq. (12),

$$C_*\big(X^J \times \Delta^J, X^J \times \partial\Delta^J\big) \xrightarrow{\hat{\mathcal{A}}_t} C_*\big(X^J\big) \otimes C_*\big(\Delta^J, \partial\Delta^J\big),$$

as the restriction is the Alexander–Whitney map obtained by setting $X = X^J$, $Y = \Delta^J$, $X_0 = \emptyset$, and $Y_0 = \partial\Delta^J$ in Eq. (2). Therefore, the restriction induces an isomorphism on homology groups and the theorem follows. □

4.4. Persistence algorithm

In this section, we give a revised persistence algorithm that computes descriptions of the representatives of homology classes, as well as barcodes. We present the algorithm over $\mathbb{Z}_2$ coefficients, but the same algorithm may be easily adapted to other fields. This is the most complete description of the algorithm and, we believe, the final form in its evolution. The algorithm works for a large class of cell complexes which we formalize after its description. To emphasize this, we assume we have a filtered cell complex throughout this section, but use the filtered simplicial complex in Fig. 11 as an example.

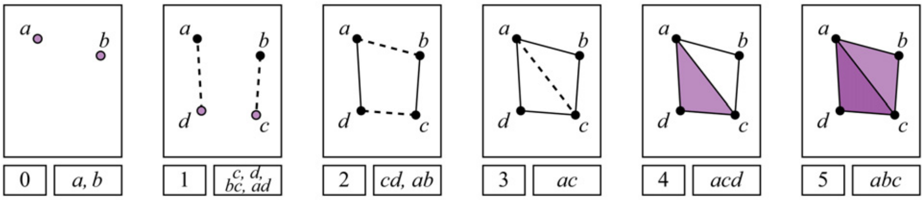


Fig. 11. A simple filtration with newly added simplices highlighted and listed.

A filtration implies a partial order on the cells. We begin by sorting cells within each time snapshot by dimension, breaking other ties arbitrarily, getting a *full order*. The algorithm, listed in pseudo-code in the procedure PAIR-CELLS, takes this full order as input. Its output consists of persistence information: It partitions the cells into *creators* and *destroyers* of homology classes, and pairs the cells that are associated to the same class. Also, the procedure computes a generator for each homology class. As we shall see, this information is stored within the *cascade* for each cell. Unlike the previous versions of the algorithm, the procedure is capable of computing generators for all homology classes, including those that never merge with the boundary class. This is the key extension described in this section.

The procedure PAIR-CELLS is incremental, processing one cell at a time. Each cell σ stores its *partner*, its paired cell, as shown in Table 1 for the filtration in Fig. 11. Once we have this pairing, we may simply read off the barcode from the filtration. For instance, since vertex b is paired with edge cd, we get interval $[0, 2)$ in the β_0-barcode. Each cell σ also stores its *cascade*, a k-chain, which is initially σ itself. The algorithm focuses the impact of σ's entry on the topology by determining whether $\partial\sigma$ is already a boundary in the complex using a call to the procedure ELIMINATE-BOUNDARIES. After the call, there are two possibilities on line 5:

1. If $\partial(\mathit{cascade}[\sigma]) = 0$, we are able to write $\partial\sigma$ as a sum of the boundary basis elements, so $\partial\sigma$ is already a $(k-1)$-boundary. But now, *cascade*$[\sigma]$ is a new k-cycle that σ completed. That is, σ creates a new homology cycle and is a *creator*.
2. If $\partial(\mathit{cascade}[\sigma]) \neq 0$, then $\partial\sigma$ becomes a boundary after we add σ, so σ destroys the homology class of its boundary and is a *destroyer*. We pair σ with the *youngest* cell τ in $\partial(\mathit{cascade}[\sigma])$, that is, the cell that has most recently entered the filtration.

During each iteration, the algorithm maintains the following invariants: It identifies the ith cell as a creator or destroyer and computes its cascade; if σ is a creator, its cascade is a generator for the homology class it creates; otherwise, the boundary of its cascade is a generator for the boundary class.

```
PAIR-CELLS(K)
1 for σ ← σ_1 to σ_n ∈ K
2   do partner[σ] ← ∅
3      cascade[σ] ← σ
4      ELIMINATE-BOUNDARIES(σ)
5      if ∂(cascade[σ]) ≠ 0
6         then τ ← YOUNGEST(∂(cascade[σ]))
7              partner[σ] ← τ
8              partner[τ] ← σ
```

The procedure ELIMINATE-BOUNDARIES corresponds to the processing of one row (or column) in Gaussian elimination. We repeatedly look at the youngest cell τ in $\partial(\mathit{cascade}[\sigma])$. If it has no partner, we are done. If it has one, then the cycle that τ created was destroyed by its partner. We then add τ's partner's cascade to σ's cascade, which has the effect of adding a boundary to $\partial(\mathit{cascade}[\sigma])$. Since we only add boundaries, we do not change homology classes.

Table 1
Data structure after running the persistence algorithm on the filtration in Fig. 11. The simplices without partners, or with partners that come after them in the full order, are creators; the others are destroyers

σ	a	b	c	d	bc	ad	cd	ab	ac	acd	abc
$partner[\sigma]$		cd	bc	ad	c	d	b	abc	acd	ac	ab
$cascade[\sigma]$	a	b	c	d	bc	ad	cd ad bc	ab cd ad bc	ac cd ad	acd	abc acd

```
ELIMINATE-BOUNDARIES(σ)
while ∂(cascade[σ]) ≠ 0
  do τ ← YOUNGEST(∂(cascade[σ]))
     if partner[τ] = ∅
        then return
        else cascade[σ] ← cascade[σ] + cascade[partner[τ]]
```

Table 1 shows the stored attributes for our filtration in Fig. 11 after the algorithm's completion. For example, tracing ELIMINATE-BOUNDARIES(cd) through the iterations of the **while** loop, we get:

#	$cascade[cd]$	τ	$partner[\tau]$
1	cd	d	ad
2	$cd + ad$	c	bc
3	$cd + ad + bc$	b	$\emptyset$

Since $partner[b] = \emptyset$ at the time of bc's entry, we pair b with cd in PAIR-CELLS upon return from this procedure, as shown in the table. The primary difference between the algorithm presented here and in prior work is that we store $cascade[\sigma]$ for all cells, instead of $\partial(cascade[\sigma])$ just for boundaries. In practice, we maintain both separately in filtration order, so the function YOUNGEST simply returns the last cell in the boundary. If we do not require generators, we just store $\partial(cascade[\sigma])$ and eliminate destroyers from the chains, giving us the prior algorithm [31].

Table 1 contains all the persistence information we require. If $\partial(cascade[\sigma]) = 0$, σ is a creator and its cascade is a representative of the homology class it created. These are the descriptions that prior versions of the algorithm could not compute and which we utilize in this paper. Specifically, the 1-cycles and 2-cycles displayed in Figs. 12 and 13 are the homology class representatives for the blowup complex, projected down (using π_X) to the base space. On the other hand, if σ is a destroyer, its cascade is a chain whose boundary is a representative of the homology class σ destroyed. This the description that was used for computing the linking numbers [10].

We end this section by formalizing the types of complexes that may serve as input to the persistence algorithm. A *persistence complex* is a family of chain complexes $\{C_{*,i}\}_{i \geqslant 0}$ together with chain map's $f_i : C_{*,i} \to C_{*,i+1}$ [31].

Definition 11 *(based persistence complex).* A *based persistence complex* is a persistence complex equipped with a choice of basis in every dimension and persistence level, such that the basis in one fixed dimension and level maps to a subset of the bases in the same dimension and higher levels under inclusion.

Note that a broad class of filtered cell complexes give rise to based persistence complexes, including simplicial complexes, simplicial sets [8,23], Δ-complexes [18], and cubical complexes [20], and blowup complexes. In each case, we can choose a standard basis of cells, where the cells are simplices, cubes, or simplicial sets.

Theorem 6 *(persistence algorithm). The procedure* PAIR-CELLS *computes persistent information for any based persistence complex.*

Proof. It is clear from the specification of the algorithm that we only use basis elements and the boundary operator for each given complex and nothing special to the geometry of the underlying complex. Therefore, our derivation and proofs in the prior paper extend [31]. □

4.5. Localization algorithm

In Section 4.3, we showed that the chain complex $C_*^{\mathcal{U}}(X)$ has the same localized homology as the simplicial blowup complex, so we may use it for computation. To compute homology using the persistence algorithm from the last section, all we need to specify is a basis for $C_*^{\mathcal{U}}(X)$ and the boundary homomorphism.

Lemma 3 *(basis). A basis for $C_k^{\mathcal{U}}(X)$ is the set composed of elements $\sigma \otimes \Delta^J$ for all $\emptyset \neq J \subseteq [n-1]$ and simplices $\sigma \in X^J$ where* $\dim \sigma + \dim \Delta^J = k$.

Proof. $C_k^{\mathcal{U}}(X)$ contains chains in $C_*^{\mathcal{U}}(X)$ with dimension k. By Definition 9, $C_*^{\mathcal{U}}(X) \subseteq C_*(X) \otimes C_*(\Delta^{n-1})$ is a sum of complexes of the form $C_*(X^J) \otimes C_*(\Delta^J)$. From the definition of the tensor product, a basis for $C_*(X) \otimes C_*(\Delta^{n-1})$ is the set of elements $\sigma \otimes \Delta^J$ with dimension equal to $\dim \sigma + \dim \Delta^J$. Note that each summand $C_*(X^J) \otimes C_*(\Delta^J)$ is the subcomplex spanned by the subset of this basis consisting of elements $\sigma \otimes \Delta^J$ for which $\sigma \in X^J$. Since each summand is a subcomplex spanned by a subset of the basis, so is their sum. □

To define the boundary, we impose total orderings on the vertices of X and Δ^{n-1}.

Lemma 4 *(boundary homomorphism). Let $\sigma \otimes \Delta^J$ be a basis element for $C_k^{\mathcal{U}}(X)$. Then,*

$$\partial\big(\sigma \otimes \Delta^J\big) = \partial\sigma \otimes \Delta^J + (-1)^{\dim \sigma} \sigma \otimes \partial \Delta^J = \sum_{i=0}^{\dim \sigma} (-1)^i \, \hat{\sigma}_i \otimes \Delta^J + (-1)^{\dim \sigma} \sum_{j=0}^{\dim \Delta^J} (-1)^j \, \sigma \otimes \hat{\Delta}_j^J, \tag{13}$$

where $\hat{\sigma}_i$ indicates that the ith vertex is deleted from the sequence.

Proof. This is just the standard formula for the boundary map in $C_*(X) \otimes C_*(\Delta^{n-1})$ [18, page 273], restricted to the basis set described in Lemma 3. □

The algorithm follows from our definitions and theorems. We begin by computing the generators of the blowup complex according to Lemma 3. We represent each element $\sigma \otimes \Delta^J$ as a pair (σ, J) and we sort the set of pairs first according to the cardinality of J, and then the dimension of σ, to get the filtration in Definition 8. We then feed this filtration, along with the boundary operator in Eq. (13), to the persistence algorithm in the last section to get the barcode. According to Definition 6, the localized attributes correspond to intervals in the barcode that contain both 0 and $n-1$. Our modified persistence algorithm also generates the localized descriptions through the cascades for creator cells.

We now briefly discuss the complexity of our algorithm. For a filtration with m generators, the persistence algorithm has running time $O(m^3)$, although in practice, linear performance has been observed for many cases [30]. In our case, m is the number of generators for the blowup complex, or equivalently, the number of its cells. Clearly, the size of the blowup complex is dependent on the cover. In the worst case, all of the simplices in our space are contained within all n sets in the cover $\mathcal{U}$, that is, each simplex has *coverage n*. Then, each simplex σ generates the cell $\sigma \times \Delta^n$, a cell with 2^n faces. That is, the blowup complex *blows up* our space to be 2^n times larger and truly deserves its name. However, this is an artificial case as this cover has no geometric information and would never be used in practice. We need to reduce the coverage card J for each simplex to reduce the size of the contributed cell $\sigma \times \Delta^J$.

In general, we do not need to compute the entire blowup complex. Computing the kth homology group requires the $(k+1)$-*skeleton*, cells with dimension less than or equal to $k+1$. Since only the first d homology groups of a d-dimensional space may be non-trivial, the largest complex we build is the $(d+1)$-skeleton. For uniform coverage c, this skeleton has less than $\sum_{i=0}^{d+1} \binom{c}{i}$ cells for each simplex.

4.6. Experiments

We have implemented our algorithm as part of a library of programs for algebraic topology. Our implementation is in C++ and utilizes the *generic programming* methodology to achieve a one-to-one correspondence between

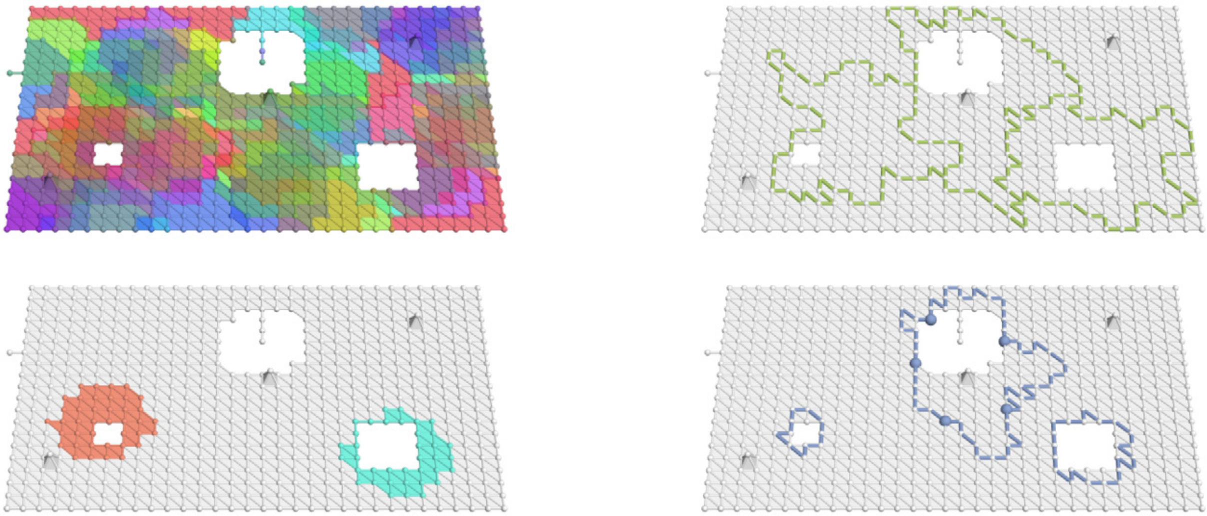


Fig. 12. Top left: A defective surface with extra edges and tetrahedra, covered by transparently colored sets based on ϵ-balls. Top right: The 1-cycles computed with homology are non-local and one goes around two holes. Bottom left: The complex highlighting the first sets in the cover that contain the small and medium holes, respectively. Bottom right: The projection (π_X) of the 1-cycles of the blowup complex localizes the two smaller holes. The large 1-cycle has portions that project onto vertices, as indicated.

the algebraic structures and their implementations [2]. We use the same code for computing homology and persistent homology of a simplicial complex and that of its blowup complex, represented as a abstract complex of simplex products. Since our focus here is on demonstrating our method, we only consider two naive methods for cover generation based on *random ϵ-balls* and *tilings*. We time our examples on a Dell PC with a 2.53 GHz P4 and 1 GB RAM running CentOS 4.3. Our examples satisfy two criteria: they are simple enough to be visualized and thoroughly understood, and their homology cannot be localized by any existing algorithm. However, our algorithm can handle complicated complexes easily as its complexity is dependent on the cover and not on the underlying space. In each case, we first compute a description using the reduction scheme for comparison.

The complex in Fig. 12 models a defective surface with 3404 simplices. This complex has extraneous edges and tetrahedra attached to it. Computing homology, we get the basis 1-cycles shown on the top right, where one cycle goes around two holes. We now use random ϵ-balls to cover our complex. The maximum coverage can be as large as the size of the cover, but the expected coverage is low. We use a ball of radius 10% the diameter of the space as our local element and take the closure of simplices that fall within a ball as a set of the cover. We try 60 random sets and place all remaining uncovered simplices in a 61st set. In the figure, we color the sets transparently and identify the two sets that contain the two smaller holes. It takes 2.60 seconds to construct the 26,627 cells of the blowup complex and 2.06 seconds to compute persistence. We show the projection of the blowup 1-cycles to the base space: our random cover localizes the two smaller holes but the not the largest since it is not contained within a single set in the cover. We note that no existing localization algorithm accepts this complex as input, as the complex is not a triangulated 2-manifold. Our algorithm handles this case without change or need for special preprocessing.

We next localize a tunnel and two voids carved out from a solid cubical block, as shown in Fig. 13. We cover the complex of 20,158 simplices using a systematic method based on tiling Euclidean spaces. The method generates covers of size at most $2^d + 1$ for complexes embedded in $\mathbb{R}^d$ (i.e. 9 in $\mathbb{R}^3$) and guarantees localization of cycles half the size of a tile. Therefore, the coverage provided here is uniform and fixed. It takes 73.56 seconds to compute the 3-skeleton with 470,484 cells and 58.54 seconds to compute persistence. The figure shows the localization relative to our cover. Note that homological descriptions for same-sized voids may be wildly different in size, but our localized descriptions are guaranteed to respect the cover. We know of no other method that works for three-dimensional complexes, or can localize two-dimensional attributes, or localize multiple types of attributes at once. Our algorithm works without change across spaces, dimensions, and covers.

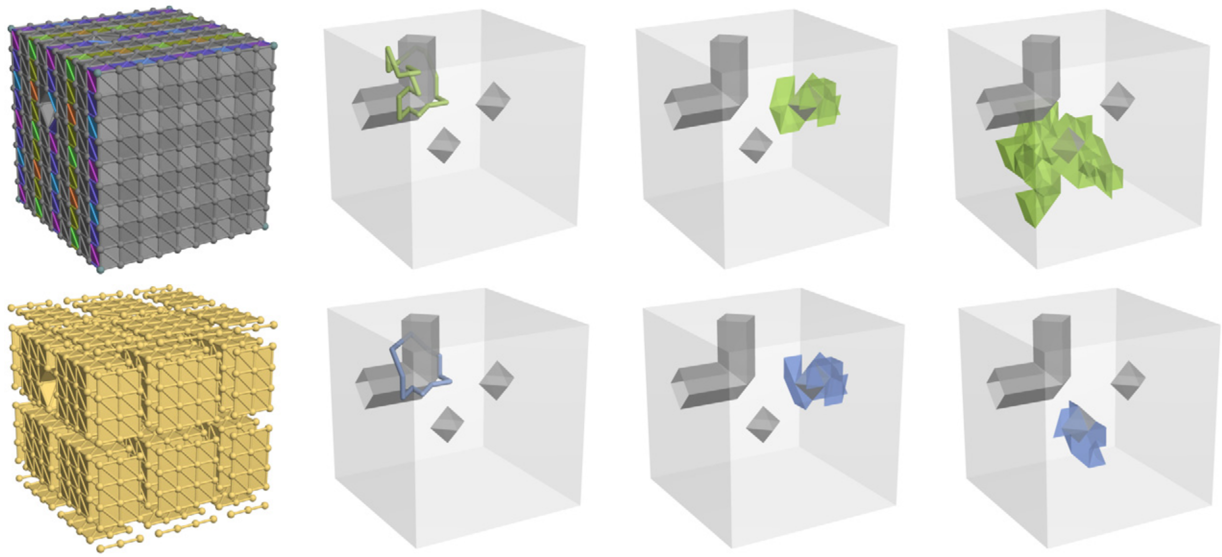


Fig. 13. We carve a tunnel and two voids from a solid cubical block. Top left: the simplicial complex representing our space, colored by the 8 sets in the cover. (For colors see the web version of this article.) Bottom left: a set in the cover has components of dimensions 1, 2, and 3. On the right, we show the descriptions on a transparent rendering of the volume. The top row renders the descriptions found by homology for the 1-cycle (tunnel) and two-cycles (voids), and the bottom row shows our localized descriptions.

5. Conclusion

In this paper, we address the problem of localization in the most general setting. Unlike previous work that focused on closed 2-manifolds, we allow for arbitrary-dimensional topological spaces. Also, we localize all topological attributes, and not just one-dimensional ones as in previous work. Any method that aims to solve the localization problem in the general case must resolve the intrinsic non-locality of topology, externalized in the Mayer–Vietoris sequence. We achieve a resolution through a new understanding of persistent homology as a computational view of functoriality. On the theory side, we begin with definitions at the singular level to be mathematical complete, but also provide provably equivalent definitions for simplicial complexes to be practical, deriving an efficient method that works at the chain complex level. On the practical side, we implement our algorithm and show results for large two- and three-dimensional complexes.

A major issue we do not address is cover construction. Including geometry through covers allows us to place arbitrary metrics on the spaces under study. We do not place any restriction on the geometry or the topology of the cover sets, as seen in our two examples, so we can tailor the cover construction to the localization requirements in different settings and even utilize multiple covers in tandem. Our approach means, however, that the localization is only as good as the cover. We do show that even naively constructed covers provide immediate information. We may therefore utilize a multilevel approach: we first use a naive cover to find an initial localization. We then recursively localize *within* each set containing an attribute to tighten the descriptions. Since we do not need to refine sets that are not topologically interesting, our recursive refinement is *adaptive* and does not need to be large. For instance, we may use hierarchical space decomposition, such as a *BSP-tree* [26] or a *kd-tree* [3], using the blowup complex descriptions at each level to guide the decomposition.

References

[1] M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, J.-C. Latombe, C. Varma, Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion, Journal of Computational Biology 10 (2003) 257–281.

[2] M.H. Austern, Generic Programming and the STL: Using and Extending the C++ Standard Template Library, Addison-Wesley, Boston, MA, 1998.

[3] J.L. Bentley, Multidimensional search trees used for associative searching, Communication of ACM 18 (1975) 509–517.

[4] K.S. Brown, Cohomology of Groups, Graduate Texts in Mathematics, vol. 87, Springer-Verlag, New York, 1982.

[5] G. Carlsson, T. Ishkhanov, V. de Silva, A. Zomorodian, On the local behavior of spaces of natural images, International Journal of Computer Vision 76 (1) (2008) 1–12.
[6] G. Carlsson, A. Zomorodian, A. Collins, L.J. Guibas, Persistence barcodes for shapes, International Journal of Shape Modeling 11 (2) (2005) 149–187.
[7] É. Colin de Verdière, F. Lazarus, Optimal system of loops on an orientable surface, Discrete & Computational Geometry 33 (3) (2005) 507–534.
[8] E.B. Curtis, Simplicial homotopy theory, Advances in Mathematics 6 (1971) 107–209.
[9] H. Edelsbrunner, D. Letscher, A. Zomorodian, Topological persistence and simplification, Discrete & Computational Geometry 28 (2002) 511–533.
[10] H. Edelsbrunner, A. Zomorodian, Computing linking numbers in a filtration, Homology, Homotopy and Applications 5 (2) (2003) 19–37.
[11] J. Erickson, Personal communication.
[12] J. Erickson, S. Har-Peled, Optimally cutting a surface into a disk, Discrete & Computational Geometry 31 (1) (2004) 37–59.
[13] J. Erickson, K. Whittlesey, Greedy optimal homotopy and homology generators, in: Proc. ACM–SIAM Symposium on Discrete Algorithms 2005, pp. 1038–1046.
[14] Q. Fang, J. Gao, L.J. Guibas, Locating and bypassing routing holes in sensor networks, in: IEEE INFOCOM, 2004, pp. 2458–2468.
[15] D. Freedman, C. Chen, Measuring and localizing homology classes, 2007, Manuscript.
[16] M.J. Greenberg, J.R. Harper, Algebraic Topology: A First Course, Mathematics Lecture Note Series, vol. 58, Benjamin/Cummings Publishing Co., Reading, MA, 1981.
[17] I. Guskov, Z. Wood, Topological noise removal, in: Graphics Interface, 2001, 19–26.
[18] A. Hatcher, Algebraic Topology, Cambridge University Press, Cambridge, UK, 2002.
[19] K. Ito, Encyclopedic Dictionary of Mathematics, second ed., MIT Press, Cambridge, MA, 1987, Ch. Complexes.
[20] T. Kaczynski, K. Mischaikow, M. Mrozek, Computational Homology, Springer-Verlag, New York, 2004.
[21] F. Lazarus, M. Pocchiola, G. Vegter, A. Verroust, Computing a canonical polygonal schema of an orientable triangulated surface, in: Proc. ACM Symposium on Computational Geometry, 2001, pp. 80–89.
[22] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, D. Fulk, The digital Michelangelo project: 3D scanning of large statues, in: Proc. SIGGRAPH, 2000, pp. 131–144.
[23] J.P. May, Simplicial Objects in Algebraic Topology, D. Van Nostrand Co., Inc., Princeton, NJ, 1967.
[24] J.R. Munkres, Topology: A First Course, Prentice-Hall Inc., Englewood Cliffs, NJ, 1975.
[25] J.R. Munkres, Elements of Algebraic Topology, Addison-Wesley, Reading, MA, 1984.
[26] B. Naylor, J. Amanatides, W. Thibault, Merging BSP trees yield polyhedral modeling results, in: Proc. SIGGRAPH, 1990, pp. 115–124.
[27] G. Segal, Classifying spaces and spectral sequences, Publications Mathématiques de l'Institut des Hautes Études Scientifiques 34 (1968) 105–112.
[28] G. Vegter, C.K. Yap, Computational complexity of combinatorial surfaces, in: Proc. ACM Symposium on Computational Geometry, 1990, pp. 102–111.
[29] Z. Wood, H. Hoppe, M. Desbrun, P. Schröder, Removing excess topology from isosurfaces, ACM Transactions on Graphics 23 (2) (2004) 190–208.
[30] A. Zomorodian, Topology for Computing, Cambridge University Press, New York, 2005.
[31] A. Zomorodian, G. Carlsson, Computing persistent homology, Discrete & Computational Geometry 33 (2) (2005) 249–274.