Localized Homology

Article · January 2008

DOI: 10.1109/SMI.2007.25 · Source: DBLP

Gunnar Carlsson
Stanford University

Localized Homology*

Afra Zomorodian
Dept. of Computer Science
Dartmouth College
Hanover, New Hampshire, USA
afra@cs.dartmouth.edu

Gunnar Carlsson
Dept. of Mathematics
Stanford University
Stanford, California, USA
gunnar@cs.stanford.edu

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 and present experiments to validate our approach in practice.

1. Introduction

Topology describes how a space is connected, reflecting the presence of certain *qualitative* features in the space, such as the tunnel in Figure 1. 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 [14] 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,

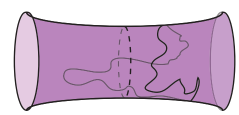


Figure 1. The localization problem

the topology of a space may have significant repercussions on the ability of geometric algorithms to perform effectively or even terminate. In *computer graphics*, undersampling and noise result in extraneous topology, such as spurious handles in reconstructed surfaces [13]. 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 [10, 18]. 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 [4]. 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 [3]. 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 analyzed a limited form of the localization problem in the past, focusing on finding tunnels within

*Research of second author partially supported by DARPA under grant HR0011-06-1-0038, and of both authors by DARPA under grant 32905.

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 [10], searching within the associated *Reeb graph* [18], cutting a surface into a disk either along the *canonical polygonal schema* [12, 17] or the *cut-graph* [8], finding a *system of loops* corresponding to a minimal presentation of the *fundamental group* of the surface [5], or obtaining a shortest set of loops that generate the fundamental group or the first homology group [9].

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 [7] such as surfaces with non-manifold structure or portions of different dimensions, such as the space in Figure 9. However, we know of no result that generalizes to localizing higher-dimensional attributes, such as enclosed volumes, or searching within higher-dimensional space, such as localizing the voids in the solid in Figure 10.

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, such as voids. We include geometry 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 [11]. A fundamental tool for this study is a *functor*, a map that not only forms algebraic images of the spaces themselves, but also of the 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 [6, 20] 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 [16]. 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. We end the paper with some preliminary experimental results where our algorithm localizes topology in spaces where other algorithms fail. Due to lack of space, we refer the reader for background and proofs to the complete version of the paper, which is available on the web.

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. Our treatment will be intuitive, as we will formalize the concepts in the next section.

2.1. Homology

Homology is the topological invariant often used in practice as it is easily computable in all dimensions. This method characterizes the connectivity of a space through the structure of its holes [11]. It extends the notion of a hole or *cycle* to all dimensions. A cycle has an intuitive meaning in $\mathbb{R}^3$: A 0-*cycle* is a connected component of the space. A 1-*cycle* is a loop that goes around a tunnel. And a 2-*cycle* is a surface that encloses an empty space. A homology cycle is really a class of equivalent *homologous* cycles, all of which characterize the same topological attribute. In each dimension k, the cycles interact to form a vector space of cycles H_k. Any basis for this vector space H_k has the same rank, the *Betti number* β_k, of the space. For example, a *torus*, the surface of a donut, has $\beta_0 = 1$ component, $\beta_1 = 2$ tunnels, and encloses $\beta_2 = 1$ void.

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 $\mathbb{R}$ [14]. Indeed, all known algorithms for computing homology are a variant of this scheme, including the *persistence algorithm* that we utilize in this paper. The key problem is that we have no geometric control over selecting the basis, and the chosen basis is often geometrically ugly and may even have multiple components. Since homology computes a basis without regard to geometry, it may choose either basis (c) or (d) for the graph in Figure 2(a) instead of the local basis (b). Since homology has no knowledge of the geometry of the space, it cannot identify "local" bases.

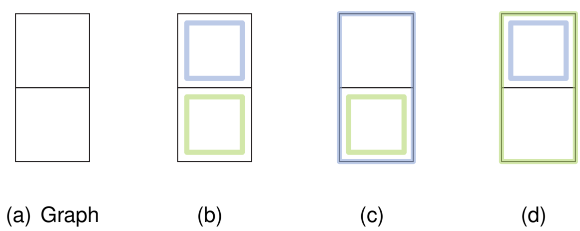


Figure 2. A graph (a) and three bases for H_1.

2.2. Adding Geometry

We wish to provide geometric knowledge to homology for selecting local bases. As a first attempt, we might compute the topology of the local pieces of the space. Suppose we *cover* the graph in Figure 2(a) with a number of local sets whose union contains the graph, as shown in Figure 3(a). Computing the homology within the sets gives us the desired local basis in Figure 2(b): Each set contains one cycle and putting the two cycles together yields the local basis.

We must be careful in assembling the cover, however, as it may overlook topological attributes. For instance, the cover in Figure 3(b) localizes only the bottom cycle. Neither set in the cover contains the cycle on the top, so the local computation fails to see it. This failure is benign, however, as it is a consequence of the cover. We may detect it easily by comparing the topology of the entire space to the result of the local computation and patching our cover.

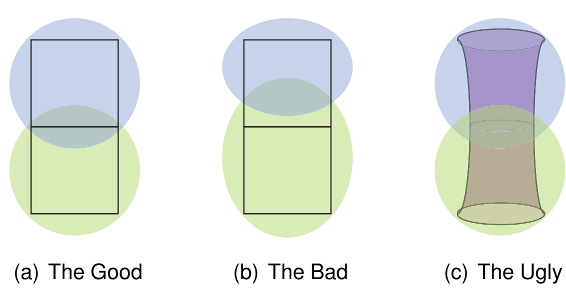


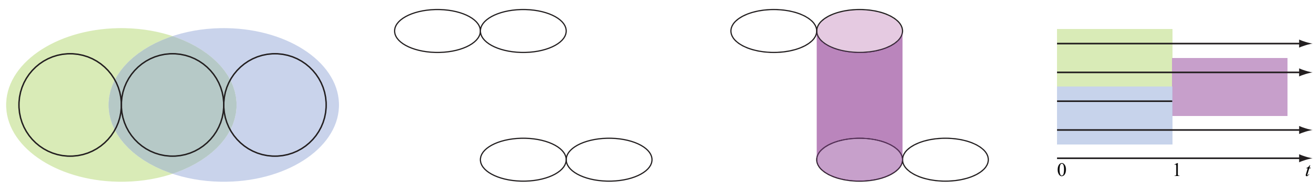
Figure 3. Covers.

A more distressing problem emerges when a topological attribute appears in multiple sets in the cover, as the tunnel in Figure 3(c). Locally, the tunnel is discovered twice, once in each set. Globally, the discovered 1-cycles are equivalent as they both go around the same tunnel. The cover effectively cuts the tunnel in two. To recover the equivalence of the two tunnels, we need to understand how the local pieces are glued to each other within the intersection of the two cover sets. That is, we need to understand the relationship of the local to the global. The theoretical gadget for exposing this relationship is the *Mayer-Vietoris sequence* [11]. Unfortunately, the sequence is only useful for computation by hand. It does not yield an implementable algorithm for arbitrary spaces. Our work may be viewed as the first attempt to make the sequence computational.

2.3. Our Approach

We begin by applying the idea from the previous section, blowing up the space into local pieces according to the cover. For example, the graph containing three cycles covered by two sets in Figure 4(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 Figure 4(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 [20]. 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 [4]. The barcode for our example, shown in Figure 4(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 Figure 4(b). At time 1, the cylinder equates the two copies of



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

Figure 4. Given a space equipped with a cover (a), we first blow up the space into local pieces (b) and then glue the pieces to get the blowup (c), giving us a filtration of two complexes at times $t = 0$ and $t = 1$. The persistence barcode (d) localizes the topology of our space with respect to the cover.

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. Localized Homology

In this section, we formalize our approach. We begin in Section 3.1 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 3.2, 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 3.3, 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 3.4. 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.

3.1. 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. Below, let $[n] = \{0, 1, \ldots, n\}$, Δ^n be the standard n-simplex, and K^n be the triangulation of Δ^n induced by the barycentric subdivision. Finally, let Δ^J be its face indexed by $J \subseteq [n]$.

Definition 1 (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 $X^J = \cap_{j\in J} X^j$ for $J \subseteq [n-1]$. The* blowup complex *$X^{\mathcal{U}} \subseteq X \times \Delta^{n-1}$ of X and $\mathcal{U}$ is*

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

$X^{\mathcal{U}}$ is equipped with two natural projection maps $\pi_X\colon X^{\mathcal{U}} \to X$ and $\pi_\Delta\colon X^{\mathcal{U}} \to \Delta^{n-1}$ given by the inclusion $X^{\mathcal{U}} \hookrightarrow X \times \Delta^{n-1}$ followed by projection onto the respective factors [16, page 108].

While the construction is standard [11, 16], there is no standard terminology, so we have adopted our own.

Example 1 (cover with two sets) Suppose X comes with cover $\mathcal{U} = \{X^0, X^1\}$ as shown on the left of Figure 5, 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 the center of the figure, where we draw Δ^1 as the interval $[0, 1]$. Following Equation (1), $X^{\mathcal{U}}$ is the union of three pieces, corresponding to the three local regions the cover defines:

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

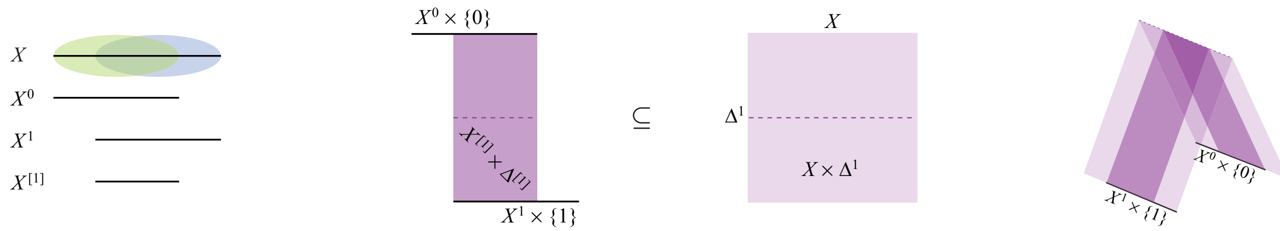


Figure 5. Blowup complex. Left: The cover $\mathcal{U} = \{X^0, X^1\}$ for space X also defines $X^{\{0,1\}} = X^{[1]}$. Center: The blowup $X^{\mathcal{U}} \subseteq X \times \Delta^1$. Right: The function f on $X^{\mathcal{U}} \subseteq X \times \Delta^1$.

Example 2 (cover with three sets) Suppose our space is a single point and our cover is three sets that contain it. Then, the blowup is Δ^2, a triangle.

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.

Proposition 1 (global) *The projection $\pi_X\colon X^{\mathcal{U}} \to X$ 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 $H_*(X^{\mathcal{U}}) \cong H_*(X)$.

Segal [16] proves the above proposition with a different hypothesis, but his result may be extended to our cases easily using standard techniques, so we omit the proof here. We include some brief remarks here for the interested specialist. Segal's result is for the case of coverings admitting a subordinate partition of unity [16, Proposition 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.

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 2 (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 $g\colon \Delta^{n-1} \to \mathbb{R}$ linearly on the complex with $g(v^J) = \dim \Delta^J$, and on Δ^{n-1} by identification. Define $f\colon X^{\mathcal{U}} \to \mathbb{R}$ by the composition $X^{\mathcal{U}} \xrightarrow{\pi_\Delta} \Delta^{n-1} \xrightarrow{g} \mathbb{R}$.*

We see f on the blowup complex on the right side of Figure 5. We filter the blowup complex using f.

Definition 3 (filtered blowup) *Let $X_t^{\mathcal{U}} = f^{-1}([0,t])$. The* filtered blowup complex *is the family $\{X_t^{\mathcal{U}}\}_{t \geq 0}$.*

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 3 (cover with three sets) In Example 2, X is a point, $\mathcal{U}$ consists of three sets that contain X, and $X^{\mathcal{U}}$ is Δ^2. Using function g, we may filter the blowup to get the filtered blowup complex $X_t^{\mathcal{U}}$, shown for the specified values in Figure 6. Note that the complex changes topologically only at integer t values.

At time 0, the blow up complex contains the local pieces of X. For Example 1, $X_0^{\mathcal{U}}$ is the two segments shown at the bottom of the hat shape in Figure 5.

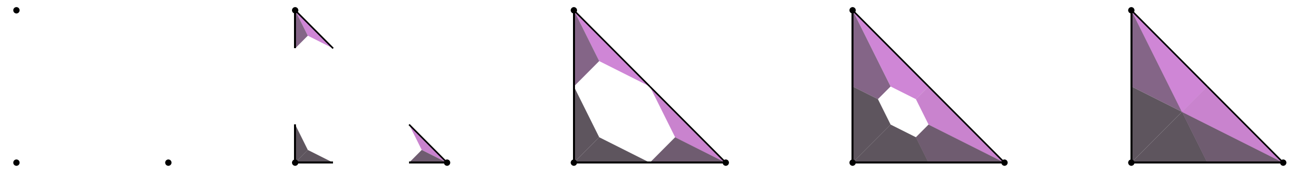
Proposition 2 (local) *The space $X_0^{\mathcal{U}}$ is the disjoint union of the local pieces of the space, i.e. $X_0^{\mathcal{U}} \approx \dot{\bigcup}_{i \in [n-1]} X^i$. Therefore,*

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

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

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, Proposition 1 asserts that $X_{n-1}^{\mathcal{U}}$ has the global homology of X. We may now state our definition for localization.

Definition 4 (localized homology) *Given a topological space X and a cover $\mathcal{U} = \{X^i\}_{i \in [n-1]}$, let $i\colon X_0^{\mathcal{U}} \hookrightarrow X_{n-1}^{\mathcal{U}}$ be inclusion, inducing $\iota_*\colon H_*(X_0^{\mathcal{U}}) \to H_*(X_{n-1}^{\mathcal{U}})$. The* localized homology of X with respect to $\mathcal{U}$ *is the image of ι_*.*



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

Figure 6. Singular blowup $X^{\mathcal{U}}_t$ for specified t for Example 3.

Applying persistent homology to the filtration, we get barcodes that describe the relationship between the local and global homology of the space. Therefore, localized homology consists of the homology classes 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 $n-1$.

The intermediate levels $X^{\mathcal{U}}_t$ also carry useful information. A cycle that is contained in the union of k sets of the cover exists in the image of $H_*(X^{\mathcal{U}}_k) \to H_*(X^{\mathcal{U}}_{n-1})$, although the converse is not always true. As such, a study of the filtered space may help refining a cover. In this paper, we only study the case $k=1$, but we hope to utilize the full filtration in the future.

3.2. Simplicial Definition

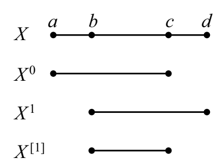
The definitions in the last section assumed that X was a topological space, so the homology groups were all *singular* homology groups rather than those attached to a simplicial complex. Singular homology examines arbitrary maps of the standard simplex and the space of such maps is an infinite dimensional space that is not computable. In this section, we modify our definitions for simplicial spaces and use simplicial homology which is finitely-generated and therefore easily computable. 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 result as the singular method. We begin by triangulating $X^{\mathcal{U}}$. Equation (1) states that $X^{\mathcal{U}}$ is a union of pieces of form $X^J \times \Delta^J$. Both terms $X^J = \cap_{j\in J} X^j$ and Δ^J are simplicial, giving us a product of simplicial complexes. Given total orderings on the vertex sets, there is a canonical way to triangulate a product space. This triangulation gives us the simplicial blowup complex. We now define both the simplicial blowup and its filtration at once.

Definition 5 (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 $X^J = \cap_{j\in J} X^j$ and*

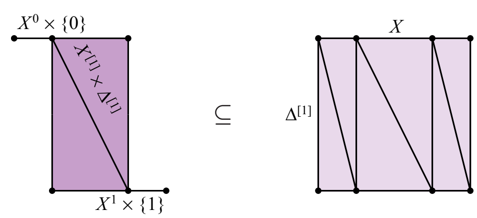
$$X^{\mathcal{U}}_t = \bigcup_{\substack{J\subseteq[n-1] \\ 0<\operatorname{card} J \le t+1}} X^J \times \Delta^J. \qquad (2)$$

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* $\pi_X \colon X^{\mathcal{U}} \to X$ *and* $\pi_\Delta \colon X^{\mathcal{U}} \to \Delta^{n-1}$. *The* filtered blowup complex *is the family* $\{X^{\mathcal{U}}_t\}_{t\ge 0}$.

Definition 5 mimics the singular definitions 2 and 3. Note that $X^{\mathcal{U}}_t$ is defined for all $t \in \mathbb{R}, t \ge 0$, but the complex changes only at integer values. Figure 7 constructs the blowup complex for Example 1 in simplicial form. The piece $X^{[1]} \times [0,1]$ is completed at time 1 in the hat shape in Figure 5, and the triangulation of the corresponding piece $X^{[1]} \times \Delta^{[1]}$ in Figure 7 also arrives at time 1.

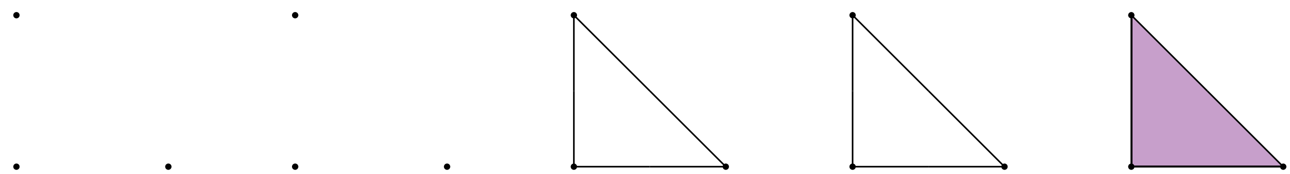


(a) Simplicial cover



(b) Blowup

Figure 7. (a) The simplicial cover $\mathcal{U} = \{X^0, X^1\} = \{ac, bd\}$ also 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

Figure 8. Simplicial blowup $X_t^{\mathcal{U}}$ for specified t for Example 4. Compare with Figure 6.

Example 4 (cover with three sets) Figure 8 shows the filtered simplicial blowup complex $X_t^{\mathcal{U}}$ for the space in Example 2 and 3. Compare with the singular blowup complex in Figure 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 $|\mathcal{U}| = \{|X^i|\}_{i\in[n-1]}$. Applying Definition 1, we get the singular blowup complex $|X|^{|\mathcal{U}|}$ that looks like the blowup in Figure 5. Clearly, the blowups are identical at integer values for t. For non-integer t, such as 0.5 in Figures 6 and 8, the two are homotopic since topology only changes at integer values. These homotopies give the equivalence of the simplicial and singular definitions, as formalized below. We provide the proof in the full version of the paper.

Proposition 3 *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 $\varphi\colon |X^{\mathcal{U}}| \to |X|^{|\mathcal{U}|}$ with restriction $\varphi_t\colon |X_t^{\mathcal{U}}| \to |X|^{|\mathcal{U}|}$ such that:*

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

Corollary 1 *The map φ of filtered spaces in Proposition 3 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 $t \geq 0$. So, their barcodes are equivalent.*

Instead of using the singular definition in the last section, we may now use the simplicial definition.

3.3. 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 Equation (2) that the filtered chain complex for the blowup complex is the family $\{C_*(X^{\mathcal{U}})_t\}_{t\geq 0}$, where $C_*(X^{\mathcal{U}})_t \subseteq C_*(X \times \Delta^{n-1})$ is

$$C_*\big(X^{\mathcal{U}}\big)_t = \sum_{\substack{J\subseteq[n-1]\\ 0<\operatorname{card} J\leq t+1}} C_*\big(X^J \times \Delta^J\big), \quad (3)$$

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 6 (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 $X^J = \cap_{j\in J} X^j$ and $C_*^{\mathcal{U}}(X)_t \subseteq C_*(X) \otimes C_*(\Delta^{n-1})$ be*

$$C_*^{\mathcal{U}}(X)_t = \sum_{\substack{J\subseteq[n-1]\\ 0<\operatorname{card} J\leq t+1}} C_*\big(X^J\big) \otimes C_*\big(\Delta^J\big). \quad (4)$$

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* $\{C_*^{\mathcal{U}}(X)_t\}_{t\geq 0}$.

The two filtered complexes defined by Equations (3) and (4) give the same localized homology.

Proposition 4 *Given simplicial space X and simplicial cover $\mathcal{U} = \{X^i\}_{i\in[n-1]}$, there is a chain map $\mathcal{A}\colon C_*(X^{\mathcal{U}}) \to C_*^{\mathcal{U}}(X)$ 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 $t \geq 0$. Consequently, the barcodes of the two chain complexes are equivalent.*

The dearth of space relegates the definition of $\mathcal{A}$ and the proof of the proposition to the full paper.

3.4. Algorithm

In the previous section, we showed that the chain complex $C_*^{\mathcal{U}}(X)$ has the same localized homology as the simplicial blowup complex. To compute the homology, we need a basis for $C_*^{\mathcal{U}}(X)$ and the boundary homomorphism. By Equation (4), $C_*^{\mathcal{U}}(X)$ is a sum of terms of the form $C_*(X^J) \otimes C_*(\Delta^J)$ over nonempty sets $J \subseteq [n-1]$. As both terms of the product are simplicial, they give rise to a canonical basis.

Proposition 5 (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 $\dim \sigma + \dim \Delta^J = k$.*

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

Proposition 6 (boundary homomorphism) *Let $\sigma \otimes \Delta^J$ be a basis element for $C_k^{\mathcal{U}}(X)$. Then,*

$$\begin{aligned} \partial\big(\sigma \otimes \Delta^J\big) &= \partial\sigma \otimes \Delta^J + (-1)^{\dim \sigma} \sigma \otimes \partial \Delta^J. \\ &= \sum_{i=0}^{\dim \sigma} (-1)^i \, \hat{\sigma}_i \otimes \Delta^J + \\ &\quad (-1)^{\dim \sigma} \sum_{j=0}^{\dim \Delta^J} (-1)^j \, \sigma \otimes \hat{\Delta}_j^J. \end{aligned} \quad (5)$$

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

The algorithm follows from our definitions and propositions. We begin by computing the generators of the blowup complex according to Proposition 5. 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 5. We then feed this filtration, along with the boundary operator in Equation (5), to the persistence algorithm [20] to get the barcode.According to Definition 4, 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.

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 [19]. 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. Clearly, we need to reduce the coverage $\operatorname{card} J$ for each simplex to reduce the size of the contributed cell $\sigma \times \Delta^J$. In the next section, we discuss the coverage for two naive cover construction schemes, one of which provides a small constant coverage that is independent of the size of the complex.

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 nontrivial, 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.

3.5. Experiments

We have implemented our algorithm as part of a library of programs for algebraic topology. 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 Figure 9 models a "defective" surface with 3,404 simplices including extraneous edges and tetrahedra. Computing homology, we get the basis 1-cycles shown in Figure 9(b), where one cycle goes around two holes. We now use random 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 in 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 blow up complex and 2.06 seconds to compute persistence. We show the projection of the blowup 1-cycles to the base space in Figure 9(d): our random cover localizes the two smaller holes but the not the largest since it is not contained within a single set of the cover. We note that no existing algorithm accepts this complex as input, as it is not a triangulated 2-manifold. Our algorithm handles

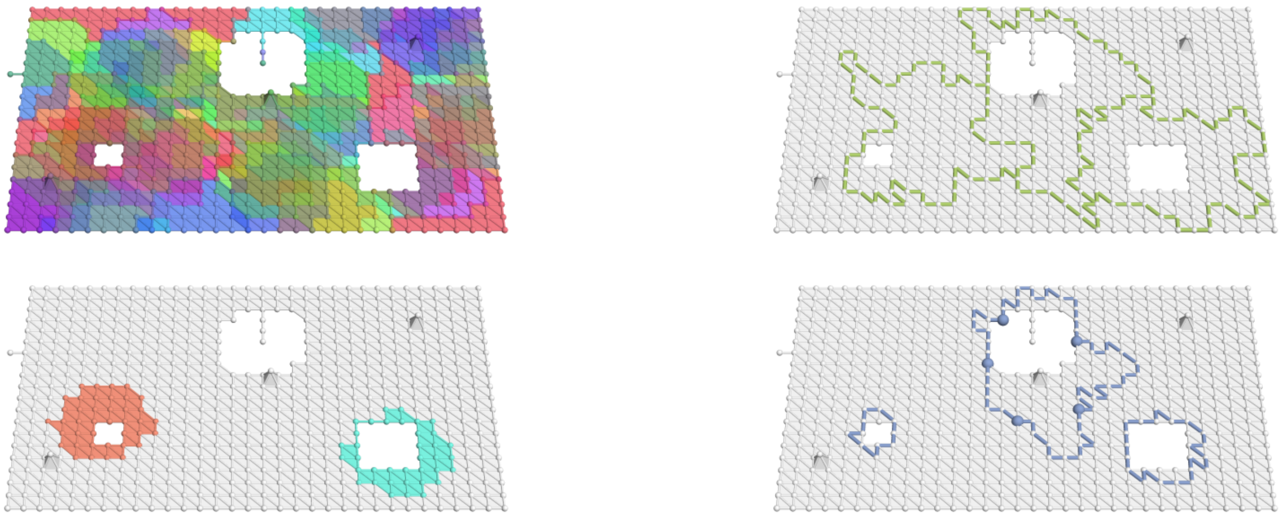


Figure 9. Top left: A defective surface with extra edges and tetrahedra, covered by transparently colored sets. Top right: The homological 1-cycles are nonlocal and one goes around two holes. Bottom left: First sets in the cover that contain the smaller holes. Bottom right: The projection (π_X) of the 1-cycles of the blowup complex localizes the smaller holes.

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 Figure 10. 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.

4. 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. 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* [15] or a *kd-tree* [2], using the blowup complex descriptions at each level to guide the decomposition.

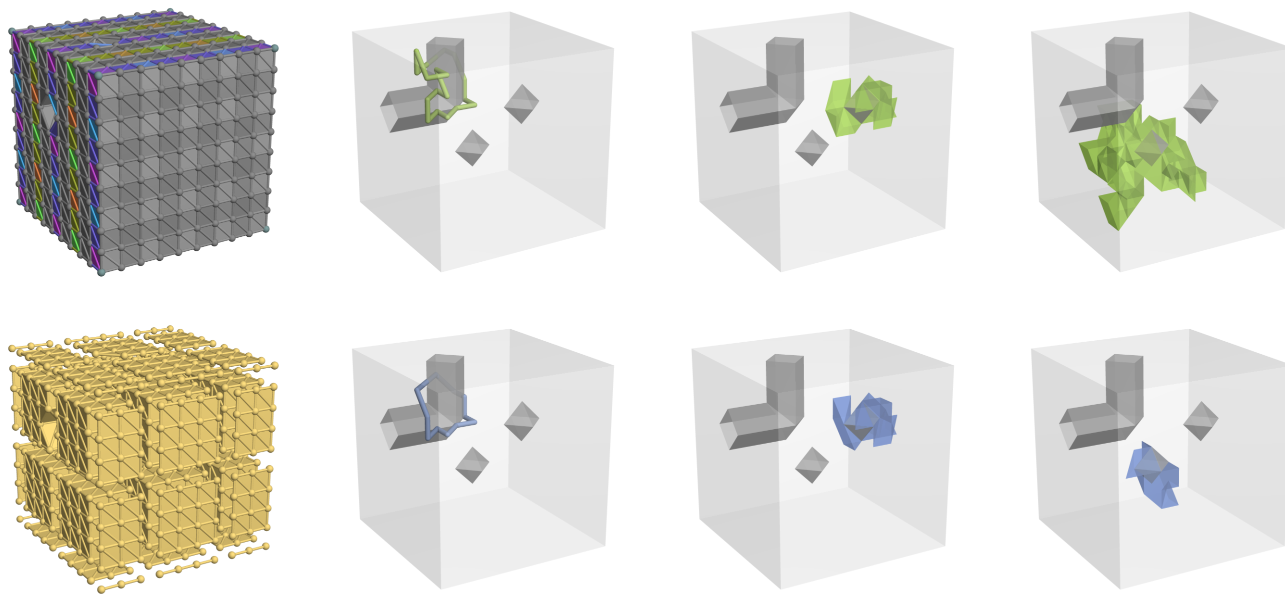


Figure 10. A tunnel and two voids carved from a solid cubical block. Top left: the simplicial complex representing the space, colored by the 8 sets in the cover. Bottom left: a set in the cover has components of dimensions 1, 2, and 3. On the right, the top row renders homological descriptions and the bottom row shows our localized descriptions.

References

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

[2] J. L. Bentley. Multidimensional search trees used for associative searching. *Comm. ACM*, 18:509–517, 1975.

[3] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 2007. To appear.

[4] G. Carlsson, A. Zomorodian, A. Collins, and L. J. Guibas. Persistence barcodes for shapes. *International Journal of Shape Modeling*, 11(2):149–187, 2005.

[5] É. Colin de Verdière and F. Lazarus. Optimal system of loops on an orientable surface. *Discrete and Computational Geometry*, 33(3):507–534, 2005.

[6] H. Edelsbrunner, D. Letscher, and A. Zomorodian. Topological persistence and simplification. *Discrete and Computational Geometry*, 28:511–533, 2002.

[7] J. Erickson. Personal communication.

[8] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. *Discrete and Computational Geometry*, 31(1):37–59, 2004.

[9] J. Erickson and K. Whittlesey. Greedy optimal homotopy and homology generators. In *Proc. 16th. Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1038–1046, 2005.

[10] I. Guskov and Z. Wood. Topological noise removal. In *Graphics Interface*, pages 19–26, 2001.

[11] A. Hatcher. *Algebraic Topology*. Cambridge University Press, Cambridge, UK, 2001.

[12] F. Lazarus, M. Pocchiola, G. Vegter, and A. Verroust. Computing a canonical polygonal schema of an orientable triangulated surface. In *Proc. 17th Annu. ACM Sympos. Comput. Geom.*, pages 80–89, 2001.

[13] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk. The digital Michelangelo project: 3D scanning of large statues. In *SIGGRAPH 00 Conference Proceedings*, pages 131–144, 2000.

[14] J. R. Munkres. *Elements of Algebraic Topology*. Addison-Wesley, Reading, MA, 1984.

[15] B. Naylor, J. Amanatides, and W. Thibault. Merging BSP trees yield polyhedral modeling results. In *SIGGRAPH 90 Conference Proceedings*, pages 115–124, 1990.

[16] G. Segal. Classifying spaces and spectral sequences. *Inst. Hautes Études Sci. Publ. Math.*, 34:105–112, 1968.

[17] G. Vegter and C. K. Yap. Computational complexity of combinatorial surfaces. In *Proc. 6th Annu. ACM Sympos. Comput. Geom.*, pages 102–111, 1990.

[18] Z. Wood, H. Hoppe, M. Desbrun, and P. Schröder. Removing excess topology from isosurfaces. *ACM Transactions on Graphics*, 23(2):190–208, Apr. 2004.

[19] A. Zomorodian. *Topology for Computing*. Cambridge University Press, New York, NY, 2005.

[20] A. Zomorodian and G. Carlsson. Computing persistent homology. *Discrete and Computational Geometry*, 33(2):249–274, 2005.