An Incremental Algorithm for Betti Numbers of Simplicial Complexes on the 3-sphere[1]

Cecil Jose A. Delfinado[2] and Herbert Edelsbrunner[2]

Abstract. A general and direct method for computing the betti numbers of a finite simplicial complex in S^d is given. This method is complete for $d \leq 3$, where versions of this method run in time $\mathrm{O}(n\alpha(n))$ and $\mathrm{O}(n)$, n the number of simplices. An implementation of the algorithm is applied to alpha shapes, which is a novel geometric modeling tool.

Key Words and Phrases. Solid modeling, geometric algorithms, graph algorithms, algebraic topology, simplicial complexes, filtrations, alpha shapes, homology groups, betti numbers, union-find, depth-first search.

---

[1]This work is supported by the National Science Foundation under grant ASC-9200301 and the Alan T. Waterman award, grant CCR-9118874. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the view of the National Science Foundation.

[2]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, Illinois 61801, USA.

1 Introduction

The main objective in the area of geometric or solid modeling is the effective specification and manipulation of geometric objects using the computer, see e.g. [10, 12]. The case where objects are imbedded in $\mathbb{R}^3$ is by far most important and has the largest number of applications. Geometric modeling is closely related to topics in topology where questions related to continuity and connectivity are studied. Given an object in $\mathbb{R}^3$, a typical topological question is how many connected components, how many independent tunnels, and how many voids there are. The number of components, tunnels, and voids have concrete interpretations. Examples are the number of heavenly bodies in a galaxy, the number of independent closed routes that go around obstacles, and the number of portions of a cell occupied by fluid.

Among the various approaches to modeling a geometric object, we consider simplicial complexes, which are collections of simplices that fit together in a natural way to form the object. In particular, we relate our findings to filtrations and alpha shapes, see e.g. [8]. The alpha shape method starts with a finite point set as input data and creates a representative family of geometric objects or shapes in an automatic and mathematically well-defined manner. Each shape is obtained from a subcomplex of the Delaunay triangulation of the point set, and the entire family of shapes forms a filtration, see section 2. This paper studies the algorithmic question of computing the topological connectivity of such objects, which is expressed in terms of betti numbers of their homology groups. In $\mathbb{R}^3$, betti numbers count connected components, independent tunnels, and voids. It is in 3 dimensions where our algorithm is complete and will probably find most of its applications.

In mathematical terminology, a geometric object is a topological space imbedded in some $\mathbb{R}^d$. Computing the homology groups of a topological space is one of the main interests in the field of algebraic topology, see e.g. [1, 9, 13, 14]. If the simplicial complex is small, then the homology group computations can be done by hand. To solve these problems in general, a classic algorithm exists and is discussed in length in [13]. It forms matrices and reduces them to a canonical form, known as the Smith normal form [15], from which one can read off the homology groups of the complex. The reduction to Smith normal form is the bottleneck of this algorithm. Starting with [11], several methods have been proposed to speed up this part of the computation. The only upper bound known on the worst-case running time of the classic reduction algorithm is double-exponential in the size of the input. However, [5] has observed that for simplicial complexes that arise in geometric design the matrices are sparse, and it is argued that in a probabilistic sense the algorithm then runs in time at most quadratic in the size of the complex.

We describe a more direct method for computing the betti numbers of the homology groups of simplicial complexes in finite dimensions. For simplicial complexes in $\mathbb{R}^3 \subset \mathbb{S}^3$, the method leads to an algorithm which runs in time $O(n\alpha(n))$, where n is the number of simplices and $\alpha(n)$ is the extremely slowly growing inverse of the Ackermann function. If the complex is represented so that the simplices incident to a given simplex can be accessed in constant time each then this can be improved to time $O(n)$. To ease the discussions, we initially deal with simplicial subcomplexes of a triangulation of $\mathbb{S}^3$. Later, we show how our algorithm extends to complexes imbedded in $\mathbb{S}^3$ without any structural assumption on the part of $\mathbb{S}^3$ not covered by the complex. Section 2 reviews some concepts from algebraic topology. Sections 3, 4, 5 and 6 develop the algorithm with an analysis and argument to its correctness. Section 7 applies the algorithm to the alpha shape approach to solid modeling. [4] is a more technical version of this paper. In that version, our method's rigorous proof of correctness using Mayer-Vietoris sequences appears.

2 Concepts in Algebraic Topology

The algorithms presented in this paper are reasonably intuitive and can easily be understood without too many technical definitions. However, a better appreciation and a deeper understanding could be attained if the reader is familiar with the concepts from homology theory presented in this section. Our terminology follows the one in [13].

Simplices, simplicial complexes, and triangulations. For $0 \leq k \leq d$, a *k-simplex* σ in R^d is the convex hull of a set T of $k+1$ affinely independent points. The *dimension* of σ is $\dim \sigma = |T| - 1 = k$. For every non-empty $U \subseteq T$, the simplex σ' defined by U is a *face* of σ. We say that σ' and σ are *incident* if σ' is a face of σ. In three dimensions we use the terms *vertex* for 0-simplex, *edge* for 1-simplex, *triangle* for 2-simplex, and *tetrahedron* for 3-simplex.

A collection of simplices, $\mathcal{K}$, is a *simplicial complex* if it satisfies two properties, namely (i) if σ' is a face of σ and $\sigma \in \mathcal{K}$ then $\sigma' \in \mathcal{K}$, and (ii) if $\sigma_1, \sigma_2 \in \mathcal{K}$ then $\sigma_1 \cap \sigma_2$ is either empty or a face of both. The largest dimension of any simplex in $\mathcal{K}$ is the *dimension* of $\mathcal{K}$. All simplices in this paper have finite dimension, and all complexes are finite collections of simplices. The *underlying space* of $\mathcal{K}$, denoted $|\mathcal{K}|$, is the set of all points in R^d contained in at least one simplex of $\mathcal{K}$. A subset $\mathcal{L} \subseteq \mathcal{K}$ is a *subcomplex* of $\mathcal{K}$ if it is a simplicial complex itself. It is *proper* if $\mathcal{L} \neq \mathcal{K}$. A particular subcomplex of $\mathcal{K}$ is its *k-skeleton* $\mathcal{K}^{(k)} = \{\sigma \in \mathcal{K} \mid \dim \sigma \leq k\}$. For example, the 1-skeleton of $\mathcal{K}$ is a simple graph in R^d. The *components* of $\mathcal{K}$ are the equivalence classes of the transitive closure of the incidence relation. $\mathcal{K}$ is *connected* if it has only one component. Since $\mathcal{K}$ is a simplicial complex it is connected iff $\mathcal{K}^{(1)}$ is connected.

An *imbedding* of a topological space A in another such space B is a continuous one-to-one map from A to B so that its inverse, restricted to the image, is also continuous. It is a *homeomorphism* if the map is also onto, that is, B is its image. A and B are *homeomorphic* if there is a homeomorphism between A and B. A *triangulation* of B is a simplicial complex, $\mathcal{T}$, whose underlying space, $|\mathcal{T}|$, is homeomorphic to B. Clearly, the underlying space of every subcomplex of $\mathcal{T}$ has an imbedding in B. We say the subcomplex is *imbeddable* in B. The d-sphere, S^d, is the set of points $x \in \mathrm{R}^{d+1}$ with unit Euclidean distance from the origin. $\mathcal{K}$ is a proper subcomplex of a triangulation of S^d if and only if it is imbeddable in R^d, e.g. by stereographic projection.

Boundary, cycles, and homology. Each k-simplex of a simplicial complex $\mathcal{K}$ can be *oriented* by assigning a linear ordering on its vertices, denoted $\sigma = [u_0, u_1, \ldots, u_k]$. Two orientations are the *same* if one sequence differs from the other by an even number of transpositions. The *boundary* of σ is

$$\partial_k \sigma = \sum_{i=0}^{k} (-1)^i [u_0, u_1, \ldots, \hat{u}_i, \ldots, u_k],$$

where the hat means that u_i is omitted. If σ is a vertex then $\partial_0 \sigma = 0$. So ∂_k maps each oriented k-simplex to a formal sum of oriented $(k-1)$-simplices. A formal sum of integer multiples of oriented k-simplices is called a *k-chain*. If the coefficient of a k-simplex in some k-chain is non-zero we say the simplex *belongs* to the chain. The group of k-chains is denoted as $\mathrm{C}_k = \mathrm{C}_k(\mathcal{K})$. The map ∂_k naturally extends to the *boundary homomorphism* $\partial_k : \mathrm{C}_k \to \mathrm{C}_{k-1}$ defined by

$$\partial_k(\sum_j a_j \sigma_j) = \sum_j a_j \partial_k \sigma_j,$$

where the a_j are integers and the σ_j are k-simplices of $\mathcal{K}$. Note that a $(k-1)$-simplex can occur in more than one term of this sum, and these terms are combined by adding the coefficients.

A number of interesting groups can be defined using the boundary homomorphism. The group of *k-cycles*, $\mathrm{Z}_k = \mathrm{Z}_k(\mathcal{K})$, is the kernel of ∂_k, that is, the subgroup of k-chains $z \in \mathrm{C}_k$ with $\partial_k z = 0$. The group of *k-boundaries*, $\mathrm{B}_k = \mathrm{B}_k(\mathcal{K})$, is the image of ∂_{k+1}, that is, the subgroup of k-chains $z \in \mathrm{C}_k$ for which there exists a $(k+1)$-chain $z' \in \mathrm{C}_{k+1}$ with $z = \partial_{k+1} z'$. It can be shown that $\partial_k \partial_{k+1} z' = 0$ for every $(k+1)$-chain z', so B_k is a subgroup of Z_k. A k-cycle $z \in \mathrm{Z}_k$ *bounds* if it is also in B_k. Finally, the quotient group $\mathrm{H}_k = \mathrm{H}_k(\mathcal{K}) = \mathrm{Z}_k | \mathrm{B}_k$ is the *k-th homology group* of $\mathcal{K}$. Each element of H_k is a class of *homologous* k-cycles. In other words, we treat two cycles the same in H_k if they differ by a cycle that bounds. Therefore, we can interpret H_k as a measure of the frequency of k-cycles that are not k-boundaries.

It is known, see e.g. [13] and [1, chapter X, §1], that $\mathrm{H}_k(\mathcal{K})$ is isomorphic to a direct sum

$$\mathrm{Z}^{\beta_k} = \mathrm{Z} \oplus \mathrm{Z} \oplus \ldots \oplus \mathrm{Z}$$

with $\beta_k \geq 0$ copies of Z, whenever $\mathcal{K}$ is imbeddable in S^3. Then $\beta_k = \beta_k(\mathcal{K}) = \beta(\mathrm{H}_k)$ is called the *k-th betti number* of $\mathcal{K}$.

The homology groups are therefore determined up to isomorphism by their betti numbers. The k-th betti number is the maximal number of independent k-cycles that do not bound. Intuitively, the 0-th betti number, β_0, is the number of components or connected pieces, the 1-st betti number, β_1, is the number of independent "tunnels", and the 2-nd betti number, β_2, is the number of "voids" or closed hollow three dimensional spaces of $|\mathcal{K}|$.

Examples. Consider the simplicial complex $\mathcal{K}$ determined by all faces of a tetrahedron. The dimension of $\mathcal{K}$ is three which is the dimension of the tetrahedron. The skeleton subcomplexes of $\mathcal{K}$ are shown in figure 1. Let z be the 1-chain $[a,b]+[b,c]+[c,a]$. Because $\partial[a,b,c] = [b,c]-[a,c]+[a,b] = [a,b]+[b,c]+[c,a]$, we

$\mathcal{K}^{(0)}$	$\mathcal{K}^{(1)}$	$\mathcal{K}^{(2)}$	$\mathcal{K}^{(3)}$
$\beta_0 = 4$	$\beta_0 = 1$	$\beta_0 = 1$	$\beta_0 = 1$
$\beta_1 = 0$	$\beta_1 = 3$	$\beta_1 = 0$	$\beta_1 = 0$
$\beta_2 = 0$	$\beta_2 = 0$	$\beta_2 = 1$	$\beta_2 = 0$

Figure 1: $\mathcal{K}^{(0)}$ is the set of four points forming four components. $\mathcal{K}^{(1)}$ is the complete graph with four vertices, consisting of one component and three independent tunnels. $\mathcal{K}^{(2)}$ is $\mathcal{K}$ without the tetrahedron, forming one component, no tunnels, and one void. $\mathcal{K} = \mathcal{K}^{(3)}$ has one component but no tunnels nor voids.

say z bounds in $\mathcal{K}^{(2)}$ but it does not bound in $\mathcal{K}^{(1)}$ which contains no 2-simplices whatsoever. z bounds so it must be a 1-cycle, and indeed $\partial z = (b-a)+(c-b)+(a-c) = 0$. Therefore, $[a,b]$, $[b,c]$ and $[c,a]$ each belong to the 1-cycle z. The cycles $z_1 = [a,c]+[c,b]+[b,d]+[d,a]$ and $z_2 = [a,b]+[b,d]+[d,a]$ are homologous in $\mathrm{H}_1(\mathcal{K}^{(2)})$ because $z_2 - z_1 = z$. However, z_1 and z_2 are not homologous in $\mathrm{H}_1(\mathcal{K}^{(1)})$ because

z does not bound in $\mathcal{K}^{(1)}$. Finally, note that every 1-cycle in $\mathcal{K}^{(1)}$ is homologous to a formal sum of the independent 1-cycles z, z_2 and $z_3 = [a,d] + [d,c] + [c,a]$. Therefore, $\beta_1(\mathcal{K}^{(1)}) = 3$.

Filtrations. A *filtration* is a sequence of simplicial complexes, where every complex is a proper subcomplex of its successor. If $\sigma_1, \sigma_2, \ldots, \sigma_n$ is a sequence of simplices, then $\mathcal{K}_1, \mathcal{K}_2, \ldots, \mathcal{K}_n$, with $\mathcal{K}_i = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$, is a filtration, provided each $\mathcal{K}_i$ is a genuine simplicial complex. We call the sequence of simplices the *filter* of the filtration. Figure 2 illustrates an example filtration.

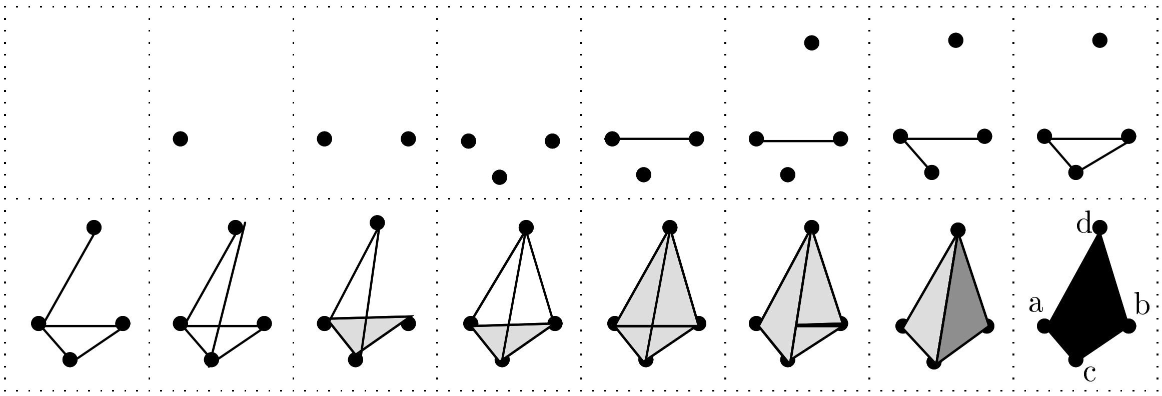

Figure 2: $\emptyset, a, b, c, ab, d, ac, bc, ad, cd, abc, bd, cd, acd, bcd, abcd$ is a filter producing the above filtration. The last complex in the corresponding filtration contains all faces of a tetrahedron.

3 The Incremental Method

The input to the method is a filter $\sigma_1, \sigma_2, \ldots, \sigma_m$ such that the simplicial complex $\mathcal{K} = \mathcal{K}_m$ is imbeddable in S^3. Only to make the exposition easier, we assume in this and the next two sections that $\mathcal{K}$ is a subcomplex of a triangulation of S^3. The incremental method processes the filter to compute the betti numbers of $\mathcal{K}$. At each step, the betti numbers of $\mathcal{K}_i$ are computed as a function of the betti numbers of $\mathcal{K}_{i-1}$, the simplex σ_i, and the complex $\mathcal{K}_{i-1}$ itself.

The largest dimension of any simplex can be at most 3, so except possibly for dimensions between 0 and $d = 3$ inclusive, all betti numbers of $\mathcal{K}$ vanish. In fact, if $\mathcal{K}$ is not a complete triangulation of S^3, $\mathcal{K}$ is imbeddable in R^3 and $\beta_3(\mathcal{K}) = 0$. To compute $\beta_\ell(\mathcal{K})$, for $0 \leq \ell \leq 3$, we use the following incremental method.

for $\ell := 0$ **to** d **do** $b_\ell := 0$ **endfor**;
for $i := 0$ **to** m **do**
    $k := \dim \sigma_i$;
    if σ_i belongs to a k-cycle of $\mathcal{K}_i$ **then** $b_k := b_k + 1$ **else** $b_{k-1} := b_{k-1} - 1$ **endif**
endfor.

Recall that σ_i belongs to a k-cycle of $\mathcal{K}_i$ if σ_i is part of a k-chain of $\mathcal{K}_i$ whose boundary is 0. Such a

k-cycle is necessarily non-bounding because no simplex in $\mathcal{K}_{i-1}$ has σ_i as a face by the filter property. If σ_i is a vertex, so $\dim \sigma_i = 0$, then σ_i belongs to a 0-cycle by definition. Hence, there will be no access to an undefined variable b_{-1}. The above method is complete if we can give a concrete procedure for deciding whether or not σ_i belongs to a k-cycle of $\mathcal{K}_i$. We will sloppily refer to this operation as "detecting a k-cycle". This will be discussed in the next section.

Correctness. All algorithms in this paper are derivatives of the incremental method. We give a somewhat informal argument for its correctness. A formal proof using Mayer-Vietoris sequences can be found in [4].

If σ_i belongs to a k-cycle of $\mathcal{K}_i$, then the addition of σ_i to $\mathcal{K}_i$ introduces a new set of k-cycles. This set is independent from the k-cycles of $\mathcal{K}_{i-1}$ because each new k-cycle contains σ_i which is absent from $\mathcal{K}_{i-1}$. Moreover, the addition of one of these new cycles to a maximal set of independent k-cycles of $\mathrm{H}_k(\mathcal{K}_{i-1})$ that do not bound forms a maximal set for $\mathrm{H}_k(\mathcal{K}_i)$. Hence, the k-th betti number increases by 1, which mirrors the action of the incremental method.

If σ_i does not belong to a k-cycle of $\mathcal{K}_i$, then no new k-cycle is introduced. However, $\mathcal{K}_i$ and $\mathcal{K}_{i-1}$ are simplicial complexes that differ only by σ_i. Therefore, the simplices belonging to $\partial\sigma_i$ are all in $\mathcal{K}_{i-1}$. Note that $\partial\sigma_i$ is a $(k-1)$-cycle in $\mathcal{K}_{i-1}$ since $\partial\partial\sigma_i = 0$. Furthermore, $\partial\sigma_i$ does not bound a k-chain in $\mathcal{K}_{i-1}$, for if it does, σ_i will belong to a k-cycle in $\mathcal{K}_i$. Hence, the addition of σ_i causes $\partial\sigma_i$ to bound in $\mathcal{K}_i$ and to become homologous to 0. Therefore, the size of the maximal set of independent $(k-1)$-cycles of $\mathrm{H}_{k-1}(\mathcal{K}_i)$ that do not bound is one less than that of $\mathrm{H}_{k-1}(\mathcal{K}_{i-1})$. That is, $\beta_{k-1}(\mathcal{K}_i) = \beta_{k-1}(\mathcal{K}_{i-1}) - 1$, which is the action taken by the incremental method.

We note that the incremental method is correct for all $d \geq 0$. Unfortunately, we can only detect 0,1 and $(d-1)$-cycles, making the method complete only for $d \leq 3$.

Remark on Euler numbers. It is interesting to note that the incremental method is sufficient to imply a classic theorem on the Euler number of a simplicial complex. Let ν_k denote the number of k-simplices of $\mathcal{K}$. The *Euler number* of $\mathcal{K}$ is defined as $\chi = \sum_{k=0}^{d}(-1)^k \nu_k$. The theorem asserts that

3.1 $\chi = \sum_{k=0}^{d}(-1)^k \beta_k$,

see e.g. [13, chapter 2, §22]. Let ν'_k be the number of k-simplices $\sigma_i \in \mathcal{K}$ so that σ_i belongs to a k-cycle of $\mathcal{K}_i$. Hence $\nu_k = \nu'_k + \nu''_k$, where ν''_k is the number of k-simplices σ_i that do not belong to any k-cycle of $\mathcal{K}_i$. Following the computations we get $\beta_k = \nu'_k - \nu''_{k+1}$. So

$$\chi = \sum_{k=0}^{d}(-1)^k \nu_k = \sum_{k=0}^{d}(-1)^k(\nu'_k + \nu''_k) = \sum_{k=0}^{d}(-1)^k(\nu'_k - \nu''_{k+1}) = \sum_{k=0}^{d}(-1)^k \beta_k$$

because $\nu''_0 = \nu''_{d+1} = 0$.

4 Supporting Data Structures

Vertices trivially belong to 0-cycles, so they do not require any data structure support to distinguish between cases. We have good data structures for detecting 1-cycles, and for detecting $(d-1)$-cycles when the simplicial complex of interest is a subcomplex of a triangulation of S^d. For $d \leq 3$ we thus can cover all cases and get an efficient algorithm. We first discuss 1-cycles. The solution for 2-cycles is similar and can be extended to detecting $(d-1)$-cycles in complexes imbedded in S^d.

Detecting 1-cycles. Let σ_i be a 1-simplex. It belongs to a 1-cycle of $\mathcal{K}_i$ iff it belongs to a 1-cycle of $\mathcal{K}_i^{(1)}$. $\mathcal{K}_i^{(1)}$ is a graph, and various efficient methods for detecting 1-cycles (cycles) in graphs are known, see e.g. [3]. For completeness we describe the method that fits best into our framework. It is based on a data structure for the so-called union-find problem.

A *union-find* data structure represents a collection of *elements* partitioned into a system of pairwise disjoint *sets*. It supports the following types of operations.

ADD(u):	Add u as the only element of a singleton set, $\{u\}$, to the system.
FIND(u):	Determine and return (the name of) the set that contains u.
UNION(A, B):	Replace the sets A and B by their union, $A \cup B$.

In our application, the elements are the vertices of the 1-skeleton and the sets correspond to its components. Initially, the system is empty. The union-find structure needs to be updated only if σ_i is a vertex or an edge. Assume the union-find structure represents $\mathcal{K}_{i-1}^{(1)}$, and consider the next simplex, σ_i. If σ_i is a vertex then ADD(σ_i) adds it to the system. If σ_i is an edge connecting vertices u and v then we find the corresponding sets, $A :=$ FIND(u) and $B :=$ FIND(v). If $A = B$ then u and v belong to the same component of $\mathcal{K}_{i-1}^{(1)}$ and thus σ_i belongs to a 1-cycle. Otherwise, σ_i does not belong to any 1-cycle in $\mathcal{K}_i^{(1)}$. Rather it connects two components which must now be merged. This is done by calling UNION(A, B).

Detecting 2-cycles. To detect 2-cycles we assume that $\mathcal{K}$ is a subcomplex of a triangulation $\mathcal{T} = \{\sigma_1, \sigma_2, \ldots, \sigma_m, \ldots, \sigma_n\}$ of S^3, and that the ordering of the simplices forms a filter. For $0 \leq i \leq n$, define $\bar{\mathcal{K}}_i = \mathcal{T} - \mathcal{K}_i$. Note that $\bar{\mathcal{K}}_i$ is in general not a simplicial complex. However, it satisfies the reverse of property (i), namely (i') if σ' is a face of σ and $\sigma' \in \mathcal{K}$ then $\sigma \in \mathcal{K}$. This can be used to characterize the 2-cycles of $\mathcal{K}_i$ in terms of the components of a graph. Let V be the set of tetrahedra of $\bar{\mathcal{K}}_i$, and let E be the set of pairs of tetrahedra, $\{a, b\}$, so that $a \cap b$ is a triangle in $\bar{\mathcal{K}}_i$. The graph $\mathcal{G}_i$ with node set V and arc set E is termed the *dual graph* of $\bar{\mathcal{K}}_i$.

Let σ_i be a triangle. σ_i belongs to a 2-cycle in $\mathcal{K}_i$ iff $\mathcal{G}_i$ has one more component than $\mathcal{G}_{i-1}$. So this means that σ_i belongs to a 2-cycle in $\mathcal{K}_i$ iff it does not belong to a 1-cycle in $\mathcal{G}_{i-1}$. Adding a simplex to $\mathcal{K}_{i-1}$ means removing the same simplex from $\bar{\mathcal{K}}_{i-1}$. Hence it appears that the dual graph must be maintained through a sequence of node and arc removals, which is computationally more expensive than a similar sequence of node and arc insertions. For this reason we reverse the processing order of the filter and obtain the empty complex by starting with $\mathcal{T}$ and removing a simplex at a time. This is done only for detecting 2-cycles and does not affect other computations.

The data structure used to represent $\mathcal{G}_i$, and thus $\bar{\mathcal{K}}_i$, is again a union-find structure. Its elements are the nodes of $\mathcal{G}_i$ (the tetrahedra of $\bar{\mathcal{K}}_i$), and the sets in the system represent the components of $\mathcal{G}_i$. Initially, $i = n$, $\mathcal{G}_n = (\emptyset, \emptyset)$, and the system that represents $\mathcal{G}_n$ is empty. The representation of $\mathcal{G}_0$ (the dual graph of $\mathcal{T} = \mathcal{T} - \emptyset$) is built by processing the simplices σ_n down to σ_1. Of course, only tetrahedra and triangles have any effect on the data structure.

To go from $\mathcal{G}_i$ to $\mathcal{G}_{i-1}$ we add the simplex σ_i to $\bar{\mathcal{K}}_i$. If σ_i is a tetrahedron then ADD(σ_i) adds it to the system. In the forward direction this corresponds to removing an isolated node of $\mathcal{G}_{i-1}$ to obtain $\mathcal{G}_i$. If σ_i is a triangle then an arc connecting the two incident tetrahedra is added to $\mathcal{G}_i$, resulting in $\mathcal{G}_{i-1}$. Using two FIND operations, we can test whether or not the two tetrahedra belong to the same component of $\mathcal{G}_i$. If they do then no further action is required. Otherwise, the two tetrahedra belong to two different components of $\mathcal{G}_i$, represented by two sets $A \neq B$ in the system. These two sets are merged by calling UNION(A, B). In the forward direction this corresponds to splitting a component. The communication between the main algorithm, which runs forward, and the 2-cycle detection mechanism, which runs backward, is based on marks left with triangles σ_i that belong to a 2-cycle of $\mathcal{K}_i$. These are the triangles that cause the execution of a UNION operation.

5 Algorithm Details

After establishing the ingredients in sections 3 and 4, we put things together to obtain the algorithm in complete detail. We do this only for $d = 3$ dimensions. Already in 4 dimensions we lack an efficient algorithm for detecting 2-cycles, and we cannot even compute the 1-st homology group because this requires detecting 1- *and* 2-cycles. For the moment, assume that the input consists of a triangulation of S^3 of which the simplicial complex of interest is a subcomplex.

The incremental algorithm. Let $\mathcal{T} = \{\sigma_1, \sigma_2, \ldots, \sigma_n\}$ be a triangulation of S^3, and for $0 \leq i \leq n$ define $\mathcal{K}_i = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$, as before. We assume that each $\mathcal{K}_i$ is a simplicial complex. The complex of interest is $\mathcal{K} = \mathcal{K}_m$, with $m \leq n$.

The first phase of the algorithm marks every simplex, σ_i, that belongs to a cycle of the same dimension in $\mathcal{K}_i$. Each vertex belongs to a 0-cycle, so all vertices get marked. To mark the appropriate edges we process the simplices in forward direction and maintain a union-find structure for $\mathcal{K}_i^{(1)}$. An edge is marked iff it does *not* cause a UNION operation. For marking the appropriate triangles we process the simplices in backward direction, from σ_n down to σ_1. A union-find structure representing the dual graph, $\mathcal{G}_i$, of $\bar{\mathcal{K}}_i = \mathcal{T} - \mathcal{K}_i$ is maintained, and a triangle is marked iff it causes a UNION operation. Finally, the only tetrahedron that belongs to a 3-cycle at the time it is processed is σ_n. This is the only tetrahedron that gets marked.

The second phase counts the marked and unmarked simplices and derives the betti numbers as simple sums of these numbers. This is done by scanning the simplices once more, in forward direction.

$b_0 := b_1 := b_2 := b_3 := 0$;
`for` $i := 1$ `to` m `do`
  $k := \dim \sigma_i$;
  `if` σ_i is marked `then` $b_k := b_k + 1$ `else` $b_{k-1} := b_{k-1} - 1$ `endif`
`endfor`.

The only case where we get $b_3 \neq 0$ is when $\mathcal{K}$ contains all tetrahedra of $\mathcal{T}$ and thus is a triangulation of S^3. If $\mathcal{K}$ is imbeddable in R^3 then $\mathcal{K} \neq \mathcal{T}$ and we can drop b_3 from the algorithm.

The analysis. The vertices, edges, triangles, and tetrahedra in $\mathcal{T}$ cause different actions in the algorithm. Let ν_k be the number of k-simplices in $\mathcal{T}$, for $0 \leq k \leq 3$. Observe that $2\nu_3 = \nu_2 \geq \nu_1 \geq 2\nu_0$. Since $n = \nu_0 + \nu_1 + \nu_2 + \nu_3$ we have $n \leq 3\nu_2$ and $n \leq 6\nu_3$, that is, at least one third of all simplices are triangles and at least one sixth of them are tetrahedra.

It is clear that phase 2 of the algorithm takes only $\mathrm{O}(n)$ time. Similarly, the vertices and the last tetrahedron can be marked in time $\mathrm{O}(n)$. The forward process, which marks edges, executes a sequence of ν_0 ADD operations, $2\nu_1$ FIND operations, and at most $\nu_0 - 1$ UNION operations. Using a standard implementation of the union-find structure this takes time $\mathrm{O}((\nu_0 + \nu_1)\alpha(\nu_0, \nu_1))$, where $\alpha(x, y)$ is the extremely slowly growing inverse of Ackermann's function, see e.g. [3]. Similarly, the backward process, which marks triangles, executes ν_3 ADD operations, $2\nu_2$ FIND operations, and at most $\nu_3 - 1$ UNION operations. This takes time $\mathrm{O}((\nu_3 + \nu_2)\alpha(\nu_3, \nu_2))$. Recall the customary notation $\alpha(x) = \alpha(x, x)$.

5.1 Let $\mathcal{K}$ be a subcomplex of a triangulation $\mathcal{T}$ of S^3, with $|\mathcal{T}| = n$. The possibly non-vanishing betti numbers of $\mathcal{K}$, $\beta_0, \beta_1, \beta_2, \beta_3$, can be computed in time $\mathrm{O}(n\alpha(n))$ and storage $\mathrm{O}(n)$.

6 Algorithm Improvements

If $\mathcal{K}$ is a subcomplex of a triangulation of S^2 then no backward computation is necessary. Hence, there is no need to consider any of the simplices that do not belong to $\mathcal{K}$. The result can then be improved to time $\mathrm{O}(m\alpha(m))$ and storage $\mathrm{O}(m)$. The following improvements are possible for complexes imbedded in S^3.

Using depth-first search. Consider the case where the simplicial complex is represented by a data structure so that for a given σ the simplices incident to σ can be accessed in constant time. An example of such a data structure is the *adjacency-list representation* which is common for graphs. The nodes are elements of a linear array. An arc is given as an index pair, so the incident nodes can be found in constant time by array look-up. The arcs incident to a node are represented by a linear list whose address is stored with the node. Given a node it is thus possible to access the incident arcs in constant time per arc. For our purpose it will be sufficient to have an adjacency-list representation for $\mathcal{T}^{(1)}$, the 1-skeleton of $\mathcal{T}$, and for $\mathcal{G}_0$, the dual graph of $\bar{\mathcal{K}}_0 = \mathcal{T}$.

Depth-first search is a standard graph search method that takes constant time per arc (edge or triangle) and can distinguish between arcs that complete a cycle and arcs that connect to a new node, see e.g. [3]. Using the data structure for $\mathcal{T}^{(1)}$ we can use depth-first search to properly mark the edges of $\mathcal{T}$. Using the data structures for $\mathcal{G}_0$ we can use depth-first search to properly mark its triangles. It is important to notice that the two depth-first searches are not coordinated with each other. Indeed, to achieve $\mathrm{O}(n)$ running time the search of $\mathcal{T}^{(1)}$ needs the freedom to visit the edges in any order it pleases. Similar for $\mathcal{G}_0$. Fortunately, every sequence in which vertices precede edges, edges precede triangles, and triangles precede tetrahedra is a filter. In particular, the sequence in which the edges are ordered by how they are visited by the search of $\mathcal{T}^{(1)}$ and the triangles are ordered in reverse by how they are visited by the search of $\mathcal{G}_0$ is a filter. The betti numbers can now be computed by traversing this filter and counting marked and unmarked simplices as before. This leads to the following improvement of 5.1.

6.1 Let $\mathcal{K}$ be a subcomplex of a triangulation $\mathcal{T}$ of S^3, with $|\mathcal{T}| = n$, and assume the 1-skeleton and the dual graph of $\mathcal{T}$ are given by their adjacency-list representations. Then the betti numbers of $\mathcal{K}$ can be computed in time and storage $\mathrm{O}(n)$.

Remark. The improvement with depth-first search sacrifices the ability to prescribe the order of the simplices. This ability is crucial for our application to alpha shapes discussed in the next section.

Non-triangulated complex complement. Up until now, we have assumed that $\mathcal{K} = \mathcal{K}_m$ is a subcomplex of a triangulation of S^3. Now, we relax this requirement and assume only $\mathcal{K}$ imbedded in R^3 is given, and its complement has no explicit representation. One way to solve this problem is to first construct a compatible triangulation of the complement. We refer to [2] for an algorithm that constructs such a triangulation. What follows is a description of another solution. For this, assume that $\mathcal{K}$ is represented so that for each edge, e, the triangles incident on e can be enumerated in the order they wrap around e, in constant time per triangle. The data structure used in [8] is an example of a representation that satisfies this requirement.

Observe that the incremental algorithm, as described earlier, uses a triangulation of the complement of $\mathcal{K}$ only to mark triangles that belong to 2-cycles. To do this without using a complement triangulation, we build a multigraph $\mathcal{M}$. Each node of $\mathcal{M}$ is an ordered pair (t, s). t is a triangle of $\mathcal{K}$ and $s \in \{+, -\}$ corresponds to one of the two sides of t. (t, s) is not a node of $\mathcal{M}$ if t is incident on a tetrahedron of $\mathcal{K}$ on side s of t. A way to distinguish the sides of a triangle is its orientation. The edges of $\mathcal{K}$ are then processed. For any two consecutive triangles t' and t'' around an edge e, if t' and t'' are not incident on a common tetrahedron of $\mathcal{K}$, then (t', s') and (t'', s'') are joined by an arc of $\mathcal{M}$. The values of s' and s'' follow from the orientations of t', t'' and e. If e is incident on only one triangle t, then an arc joins $(t, +)$ and $(t, -)$. If e is incident on no triangle, no arc is introduced by e. Observe that the components of $\mathcal{M}$ correspond to connected surface pieces of $\|\mathcal{K}\|$.

The next step is to do a depth-first search of $\mathcal{M}$. With no increase in time complexity, the components of $\mathcal{M}$ can be identified. For each component, the nodes in the component can each be made to point to a representative node. This creates the sets for the union find data structure, $\mathcal{M}_m$, for $\mathcal{M}$.

We can now perform the reverse traversal of the filter, starting with σ_m. $\mathcal{M}_m$ takes the place of the union-find data structure for $\mathcal{G}_m$ described in section 4. If a tetrahedron σ_i is processed, a new set is added to $\mathcal{M}_i$ with elements (t, s), t a triangle in the boundary of σ_i and σ_i on the s side of t. When a triangle σ_i is visited, if $\text{FIND}((\sigma_i, +)) \neq \text{FIND}((\sigma_i, -))$, then $\text{UNION}((\sigma_i, +), (\sigma_i, -))$ is performed and σ_i is marked. Otherwise, σ_i is left unmarked.

After this step, the algorithm can proceed as before. Note that if only the betti numbers of $\mathcal{K}$ are of interest, then $\beta_0(\mathcal{K})$ can be computed by performing a depth-first search of the 1-skeleton, $\mathcal{K}^{(1)}$, of $\mathcal{K}$. To compute $\beta_2(\mathcal{K})$, create the graph $\mathcal{N}$. The nodes of $\mathcal{N}$ are the triangles of $\mathcal{K}$ and two nodes are connected by an arc if the triangles share an edge. Like $\mathcal{K}^{(1)}$, the number of components of $\mathcal{N}$ can be computed by performing a depth-first search. Then, it is clear that $\beta_2(\mathcal{K})$ is just the number of components of $\mathcal{M}_m$ minus the number of components of $\mathcal{N}$. Finally, $\beta_1(\mathcal{K})$ is computed using the Euler formula and the values of $\beta_0(\mathcal{K})$ and $\beta_2(\mathcal{K})$. We get the following result.

6.2 Let $\mathcal{K}$ be imbedded in R^3 and let m be the number of simplices in $\mathcal{K}$. If for each edge $e \in \mathcal{K}$, the triangles incident on e can be enumerated in the order they wrap around e in $\mathrm{O}(1)$ time per triangle, then the betti numbers of $\mathcal{K}$ can be computed in time and storage $\mathrm{O}(m)$. Also, the betti numbers of all $\mathcal{K}_i$ in the filtration can be computed in time $\mathrm{O}(m\alpha(m))$ and storage $\mathrm{O}(m)$.

7 Signatures for Alpha Shapes

The incremental algorithm is useful especially when the betti numbers of all simplicial complexes in a filtration need to be computed. A case when this happens is in the computation of the betti number signatures for alpha shapes. A comprehensive discussion of the family of α-shapes of a finite point set is beyond the scope of this paper. As a substitute we refer the reader to papers on two-dimensional [7] and three-dimensional [8] α-shapes. Computing betti numbers for α-shapes is our main motivation for developing the algorithms in this paper.

A brief description of three-dimensional α-shapes. Let S be a finite point set in R^3, with $|S| \geq 4$, and let $\mathcal{D} = \mathcal{D}(S)$ be its Delaunay triangulation, see e.g. [6]. Provided the points are in general position, $\mathcal{D}$ is indeed a simplicial complex. Commonly, it is defined so that its underlying space, $|\mathcal{D}|$, is the convex hull of S. It is more convenient for us to add a point "at infinity" and connect it to all simplices on the boundary of the convex hull of S. The resulting simplicial complex is a triangulation of S^3, which we denote by $\mathcal{T}$.

For each non-negative real α, the *α-complex* of S is a subcomplex of $\mathcal{D}$ and therefore also of $\mathcal{T}$. The *α-shape* of S is the underlying space of the α-complex. It is defined so that for $\alpha = 0$ we get the point set S itself, and for sufficiently large α we get the convex hull of S, see [8]. Although a shape is defined for every non-negative real α, there are only finitely many different subcomplexes of $\mathcal{D}$ and therefore only finitely many different α-shapes. It is convenient to index the α-shapes and α-complexes by position. Let s be the number of different α-complexes, denoted $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_s$. For $1 \leq i \leq s$, the α-shape that corresponds to the i-th α-complex is $\mathcal{S}_i = \|\mathcal{C}_i\|$. With increasing index the corresponding α-value also grows.

For convenience, define $\mathcal{C}_0 = \emptyset$ and $\mathcal{C}_{s+1} = \mathcal{T}$. The sequence $\mathcal{C}_0, \mathcal{C}_1, \ldots, \mathcal{C}_{s+1}$ of simplicial complexes is a filtration which is a scattered subsequence of the filtration $\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_n = \mathcal{T}$, where $\mathcal{K}_i = \{\sigma_1, \sigma_2, \ldots, \sigma_i\}$, see [8]. We assume that the simplices of $\mathcal{T}$ are ordered so that each $\mathcal{K}_i$ is indeed a simplicial complex. For each simplex $\sigma_i \in \mathcal{T}$, let $\lambda(i)$ be the smallest index ℓ so that $\sigma_i \in \mathcal{C}_\ell$. This means $\mathcal{C}_{\lambda(i)}$ is the smallest α-complex that contains $\mathcal{K}_i$, and $\mathcal{C}_{\lambda(i)} = \mathcal{K}_i$ iff $\lambda(i) \neq \lambda(i+1)$.

Computing Signatures. The implementation of three-dimensional α-shapes reported in [8] includes a small number of signature functions that follow the evolution of the α-shape as α increases from 0 to $+\infty$. Let $[s]$ denote the set $\{1, 2, \ldots, s\}$. By a *signature function* we mean a function $f : [s] \to R$ that maps each index $\ell \in [s]$ to a value $f(\ell)$ in some range R. For reasons of usefulness the function should be defined so that $f(\ell)$ expresses some property of the α-shape $\mathcal{S}_\ell$. For example, $f(\ell)$ could express a combinatorial property, such as the number of triangles bounding $\mathcal{S}_\ell$, or a metric property, such as the surface area of $\mathcal{S}_\ell$.

In this section we are interested in three topological signature functions that count the number of components, independent tunnels, and voids of $\mathcal{S}_\ell$. For $0 \leq k \leq 2$ define

$$\beta_k : [s] \longrightarrow \mathrm{Z}$$

so that $\beta_k(\ell)$ is the k-th betti number of $\mathcal{S}_\ell$. The homology groups of $\mathcal{S}_\ell$ and $\mathcal{C}_\ell$ are the same, so $\beta_k(\ell) = \beta(\mathrm{H}_k(\mathcal{C}_\ell))$. Each signature function, β_k, is represented by a linear array, $b_k[1..s]$.

We can now modify the algorithm of section 5 to compute the signature functions β_0, β_1, and β_2 of all α-shapes of S. Phase 1, which marks the simplices, is exactly the same as in section 5. The only change in phase 2 is that for some values of i the computed betti numbers need to be stored in the appropriate elements of the three arrays.

$b_0[1] := b_1[1] := b_2[1] := 0$;
$\ell := 1$;
`for` $i := 1$ `to` n `do`
    $k := \dim \sigma_i$;

```
        if σ_i is marked then b_k[ℓ] := b_k[ℓ] + 1 else b_{k-1}[ℓ] := b_{k-1}[ℓ] - 1 endif;
        if ℓ < s and λ(i) ≠ λ(i+1) then
          b_0[ℓ+1] := b_0[ℓ]; b_1[ℓ+1] := b_1[ℓ]; b_2[ℓ+1] := b_2[ℓ]; ℓ := ℓ + 1
        endif
    endfor.
```

Clearly, the asymptotic complexity of this algorithm is the same as of the algorithm in section 5. We thus obtain the main result of this section.

7.1 The signature functions that map $\ell \in [s]$ to the 0-th, 1-st, and 2-nd betti numbers of $\mathcal{S}_\ell$ can be computed in time $\mathrm{O}(n\alpha(n))$ and storage $\mathrm{O}(n)$.

8 Discussion

This paper presents an incremental method for computing the betti numbers of a topological space represented by a simplicial complex. It is complete and has an efficient implementation for simplicial complexes imbeddable in R^3 or S^3. The algorithm is an example of how algorithmic techniques developed for graphs can be applied and extended to complexes of dimension higher than one. It is to be hoped that this is a step towards a revived interest in algorithmic problems in algebraic topology. As demonstrated in this paper, these algorithms do not necessarily have algebraic flavor. Indeed, we see our algorithm as evidence that combinatorial algorithms can outperform algebraic methods designed to solve the same problems.

The $\mathrm{O}(n\alpha(n))$ time implementation of the three-dimensional algorithm has been coded and incorporated into a software package on alpha shapes. This is available via anonymous ftp from ftp@ncsa.uiuc.edu. Given a finite point set in R^3, the program computes signatures, displayed as graphs of one-dimensional functions, that show the betti numbers of the evolving α-shape.

The most interesting unanswered question concerns data structures that give a complete and an efficient implementation of our incremental method for simplicial complexes not imbeddable in S^3.

References

[1] P. Alexandroff and H. Hopf. *Topologie I.* Julius Springer, Berlin, 1935.

[2] M. Bern. Compatible tetrahedralizations. *In* "Proc. 9th Ann. Sympos. Comput. Geom. 1993", 281–288.

[3] T. H. Cormen, Ch. E. Leiserson and R. L. Rivest. *Introduction to Algorithms.* The MIT Press, Cambridge, Mass., 1990.

[4] C. J. A. Delfinado and H. Edelsbrunner. An incremental algorithm for betti numbers of simplicial complexes. *In* "Proc. 9th Ann. Sympos. Comput. Geom. 1993", 232–239.

[5] B. R. Donald and D. R. Chang. On the complexity of computing the homology type of a triangulation. *In* "Proc. 32nd Ann. IEEE Sympos. Found. Comput. Sci. 1991", 650–661.

[6] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, Germany, 1987.

[7] H. Edelsbrunner, D. G. Kirkpatrick and R. Seidel. On the shape of a set of points in the plane. *IEEE Trans. Inform. Theory* **IT-29** (1983), 551–559.

[8] H. Edelsbrunner and E. P. Mücke. *ACM Trans. Graphics* **13** (1994), 43–72.

[9] P. J. Giblin. *Graphs, Surfaces, and Homology.* Second edition, Chapman and Hall, London, 1981.

[10] Ch. M. Hoffmann. *Geometric and Solid Modeling, an Introduction.* Morgan Kaufmann, San Mateo, California, 1989.

[11] R. Kannan and A. Bachem. Polynomial algorithms for computing the Smith and Hermite normal forms of an integer matrix. *SIAM J. Comput.* **8** (1979), 499–507.

[12] M. Mäntylä. *An Introduction to Solid Modeling.* Computer Science Press, Rockville, Maryland, 1988.

[13] J. R. Munkres. *Elements of Algebraic Topology.* Addison-Wesley, Redwood City, California, 1984.

[14] J. J. Rotman. *An Introduction to Algebraic Topology.* Springer-Verlag, New York, 1988.

[15] H. J. Smith. On systems of indeterminate equations and congruences. *Philos. Trans.* **151** (1861), 293–326.