Signed Simplicial Decomposition and Intersection of *n*-d-Polytope Complexes

Norbert Paul, Mathias Menninghaus
Geodätisches Institut Karlsruhe (GIK), KIT
norbert.paul@kit.edu, mathias.menninghaus@kit.edu

Abstract: The spatial configuration of building spaces and elements together with their mutual topological relations can be considered a polyhedral complex. Polytope complexes are their *n*-d-generalization and a natural concept for accessing spatio-temporal data in general. Accordingly, complex intersection is the corresponding *n*-d-generalisation of map overlay in GIS and also a generalisation of polyhedral intersection—an important operation for spatial building modelling. As simplicial decomposition in general is expensive this paper presents *signed* simplicial decomposition as a more efficient alternative. Decompose the interior of an *n*-d-element into a *linear combination* of positive and negative simplices that add to or subtract from the interior. This paper presents an efficient purely combinatorial *n*-d-polytope complex decomposition algorithm and an algorithm to carry out geometric intersection of simplices. These intersections are signed, too, and sum up to a representation of the intersection complex to be computed.

1 Introduction and Related Work

An important part of Building Information Modelling (BIM) is handling spatial data. The spatial configuration of building spaces and elements together with their topology can be considered a polyhedral complex (Hatcher 2002). Polytope complexes are their *n*-d-generalization and also generalize 2-d polygon meshes used in geoinformation systems (GIS). Complex intersection is the corresponding *n*-d-generalisation of map overlay in GIS and also a generalisation of polyhedral intersection—an important operation for spatial modelling.

(Cohen et .al. 1979) describes an unsigned polytope decomposition similar to the one presented here but which is only suitable for the convex case. In general, partitioning the elements of a complex into simplices, intersecting these pair-wise, and then combining the intersections into a resulting intersection complex may raise the prohibitive costs of $O(n^4)$ because convex partitioning in general is known to be of size $\Omega(n^2)$ (Chazelle 1984).

But often *signed* decomposition is a more efficient alternative to partitioning: Divide the polytope shape into a *linear combination* of shapes that sum up to the entire volume. Positive shapes add to and negative shapes subtract from the volume. (Bulbul, et. Al. 2009) present such a singed decomposition method for polyhedra called “AHD: alternating hierarchical decomposition”. We, however, suppose that the AHD-method does not always terminate and we will present here a purely combinatorial *n*-d-polytope complex decomposition algorithm which efficiently produces signed oriented simplices. It only depends on topological information and can decompose non-convex polytopes of arbitrary dimension. Then the product of sign and orientation tells if the simplex (hyper-) volume is additive or subtractive.

Geometric *n*-d-simplex intersection is done by the active-set-method (Nocedal et al. 2006), intersection boundary computation similar to (Lefschetz 1926). The result then has some undesirable “strange” features which must be corrected to get the final intersection complex.

2 Basic Notions

A building model is composed of spatial objects each surrounded by a boundary. Such objects may share a common boundary element in which case one object is at one side of that element and the other object on its other side. A surface element is oriented by specifying one side of it "positive" and name the other one its "negative" side. Volume-face incidences can be signed accordingly. A volume's surface (or boundary) consists of faces and each face has boundary edges. When an edge is orientated then, with respect to the volume, such a common edge has one face to its "left" and one face to its "right".

Now consider an edge at the surface of a volume. That edge can have an arbitrary number of incident faces but only two of them touch this volume: One to the left and one to the right when looking from outside. Now consider the left-hand-sided face. When the edge orientation is compatible with the face orientation the edge-face incidence gets a value of +1. We then see the front, the volume we look at is at the rear side, and its volume-face-incidence value is, say, –1. If we multiply these incidence values along the volume-face-edge incidence path we get –1. But if that face orientation is incompatible with the edge orientation we see its rear side. Then *both* incidence values are inverted and so their product still is –1, independent of the orientation of the face. The reader may verify that the right-hand-sided face always yields a volume-face-edge incidence path value of +1 independent of the orientation of the face. Then the sum of all these "path values" between each incident face-edge-pair—the boundary of the volume booundary—is zero. The same is also true for all incident face-vertex pairs.

Now we describe a formal "trick" from algebraic topology: We consider each vertex a base vector of a vector space. For example, a, b, and c be three distinct vertices of some vertex set C_0. Linear combinations like $2a + b - c$ are then considered "vectors" with an "a-coordinate" of +2, a b-coordinate of +1, a c-coordinate of –1, and the v-coordinate of every other vertex $v \in C_0$ is zero. Note that this addition is only formal and must not be confounded with the vector sum of the vertex coordinates. The same formal trick is done with the other element sets like edges, faces, and volumes: Each such set C_i of the i-dimensional elements is assigned such a formal vector space $\mathbb{Z}C_i$ of dimension $|C_i|$. When we arbitrarily orient each element the boundary of each $(i+1)$–element c is a linear combination of i-elements: $\partial(c) = \alpha_1 \cdot b_1 + \alpha_2 \cdot b_2 + \alpha_3 \cdot b_3 + \ldots + \alpha_n \cdot b_n$. Here the b_j are the *i-elements* and α_j the *boundary coefficients*. α_j is +1 if the orientation of b_j is "compatible" with that of c, –1 on "contrary" orientation, and zero if c and b_j are not incident. The following example illustrates this: Two faces A and B are bounded by edges, and each edge is bounded by two vertices.

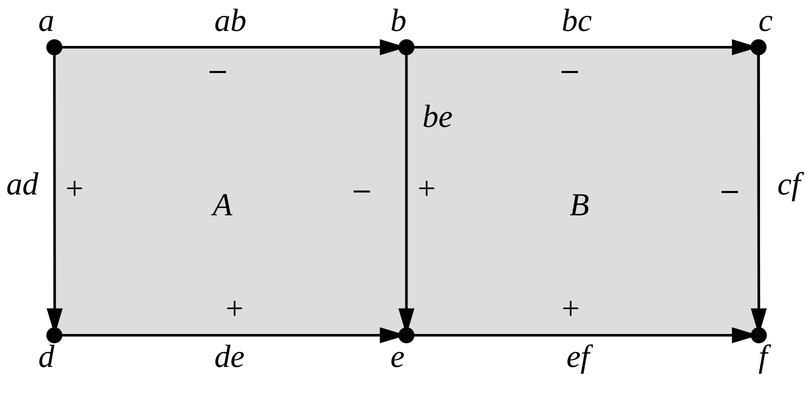

Image 1: A complex of two adjacent rectangles A and B. Note how the signs of the face-edge-incidences reflect their being counter-clockwise or not.

The orientation of each edge named *ab* shall be: “*ab* runs from *a* to *b*” which we denote by $\partial(ab) = b - a$, and the face orientations should all be “counter-clockwise”. (Hatcher 2002) gives a precise definition of “orientation”. Then the boundaries of *A* and *B* are

$$\partial(A) = ad + de - be - ab, \qquad \text{and} \qquad \partial(B) = be + ef - cf - bc.$$

The sign of a face-edge-incidence is positive if the edge runs counter-clock-wise along the face’s boundary and negative otherwise. Composing the linear boundary function ∂ we get $\partial\partial(A) := \partial\circ\partial(A) = \partial(\partial(A)) = 0$ and $\partial\partial(B) = 0$. We only demonstrate the case $\partial\partial(A)$:

$$\begin{aligned}\partial\partial(A) &= \partial(\partial(A)) = \partial(ad + de - be - ab) = \partial(ad) + \partial(de) - \partial(be) - \partial(ab)\\ &= (d - a) + (e - d) - (e - b) - (b - a)\\ &= d - a + e - d - e + b - b + a = 0.\end{aligned}$$

Definition (Algebraic Complex): *A sequence $C_n, C_{n-1}, \dots, C_0$ of n Abelian groups (such as vector spaces) together with a sequence $\partial_{n-1}, \partial_{n-2}, \dots, \partial_0$ of group homomorphisms (such as linear function) $\partial_i : C_{i+1} \to C_i$, is called an* algebraic complex*, denoted by*

$$C : C_n \xrightarrow{\partial_{n-1}} C_{n-1} \xrightarrow{\partial_{n-2}} \cdots \xrightarrow{\partial_0} C_0\,,$$

iff the compositions $\partial_{i-1}\circ\partial_i$ $(i = 0, \dots, n-1)$ of every pair of consecutive homomorphisms is trivial (such as the constant zero function). The functions ∂_i are then called the boundary operators *of C. The number n of these boundary operators is then called the* dimension *of the complex.*

The boundary coefficients α_b for each boundary $\partial(c) = \sum_{b\in C_i} \alpha_b \cdot b$ of every element *c* can be stored into a sparse matrix with row index *c* and column index a_j. We will often make a difference between a weaker “algebraic” equality and a stronger “combinatorial” equality. *Combinatorially* $e - d \neq e - d + 0\cdot a$ holds because $0\cdot a$ is missing on the left side, but *algebraically* both sides are equal. Combinatorial discrimination facilitates storage of topological information as, for example, by the following abstract relational database schema:

Definition (Relational Complex): *A sequence $E_n, E_{n-1}, \dots, E_0$ of entity types together with a sequence $R_{n-1}, R_{n-2}, \dots, R_0$ of relation types $R_i :\subseteq E_{i+1} \times E_i \to \mathbb{Z}$ is called a* relational complex*, iff the “matrix product” $R_i \cdot R_{i-1}$ $(i = 0, \dots, n-1)$ of every pair of consecutive relations, considered sparse $E_{i+1} \times E_i$ integer matrices, only contains zero entries. The relations R_i are then called the* incidence matrices *of the complex. The number n of incidence matrices is the* dimension *of the relational complex.*

Table 1: A relational complex for the example on Image 1. Keys are underlined.

Face
<u>id</u>
A
B

FaceBoundary		
<u>face</u>	<u>edge</u>	alpha
A	*ab*	–1
A	*ad*	+1
A	*be*	–1
A	*de*	+1
B	*bc*	–1
B	*be*	+1
B	*cf*	–1
B	*ef*	+1

Edge
<u>id</u>
ab
ad
bc
be
cf
de
ef

EdgeBoundary		
<u>edge</u>	<u>vertex</u>	alpha
ab	*a*	–1
ab	*b*	+1
ad	*a*	–1
ad	*d*	+1
	…	
ef	*e*	–1
ef	*f*	+1

Vertex		
<u>id</u>	x	y
a	0.0	1.0
b	1.0	1.0
c	2.0	1.0
d	0.0	0.0
e	1.0	0.0
f	2.0	0.0

The above table presents a 2-d relational complex for our example complex from Image 1 with some geometric vertex coordinates added. The matrix product of the boundary tables can be computed with SQL by

```
select FB.face, EB.vertex, sum(FB.alpha * EB.alpha) as alpha
from FaceBoundary FB, EdgeBoundary EB
where FB.edge = EB.edge
group by FB.face, EB.vertex;
```

It may seem somewhat verbose to store an edge boundary by such "signed" foreign key references instead of using the more common relational schema

```
Edge(id, start:Vertex, end:Vertex)
```

but the added flexibility to store "strange" edges will turn out to be useful later. Now a relational complex should represent a geometric object:

Definition (Polytope Complex)*: A relational complex* $C : C_n \xrightarrow{R_{n-1}} C_{n-1} \xrightarrow{R_{n-2}} \cdots \xrightarrow{R_0} C_o$ *is called* polytope complex, *iff the elements in* C_i *represent disjoint i-dimensional oriented connected flat manifolds in a real vector space* $\mathbb{R}^m$ *with* $m \geq n$.

This generalizes GIS-polygon topologies and boundary representation models. Now we state our problem: Given two polytope complexes A and B which represent a partitioning of some shape in $\mathbb{R}^n$ presented as relational complexes. Then find a complex $A \cap B$ of the non-empty intersections of the elements in A and B.

3 Signed Boundary Decomposition

The idea how to solve our problem is quite obvious: disassemble each element into parts that are easy to intersect and intersect them. Mostly "easy to intersect" means "convex" and convex partitioning is expensive in general (Chazelle 1984). So our alternative is to compute "positive" and "negative" convex parts that sum up to the partitioned element. The intersection of two such signed parts is positive, iff both parts have the same sign and it is negative if the signs differ. We will now decompose a (relational) polytope complex into simplices such that the decomposition function and the boundary operators "match well". But first we have to introduce another notion. When from all possible variants of "simplicial complex" none suits our needs we define our own:

Definition (Simplex, Simplicial Complex) *An* oriented abstract n-simplex *is a tuple* $\langle v_0, v_1, \ldots, v_n \rangle$ *of* $n + 1$ *distinct elements. An* oriented abstract simplicial complex *is a pair* $(K, <)$ *where K is a finite abstract simplicial complex and* $<$ *is a total ordering on all its vertices. A tuple* $\langle v_0, \ldots, v_n \rangle$ *is a* simplex of $(K, <)$, *iff* $\{v_0, \ldots, v_n\}$ *is an abstract simplex in K and* $v_0 < v_1 < \ldots < v_n$ *holds.*

From now on, when we say "simplicial complex" we refer to the above. We remind that the boundary of a simplex $\langle v_0, v_1, \ldots, v_n \rangle$ is defined by "alternatingly signed vertex removals" as

$$\delta\langle v_0, v_1, \ldots, v_n \rangle := +\langle v_1, \ldots, v_n \rangle - \langle v_0, v_2, \ldots, v_n \rangle + \ldots \pm \langle v_0, v_1, \ldots, v_{n-1} \rangle.$$

Our signed simplicial decomposition is a linear map that assigns each n-d-element in our input complex C a linear combination of oriented n-simplices made up of the vertices in C_0. The boundary of the decomposition is then (algebraically) equal to the decomposition of the boundary. Such maps are called *complex morphisms*.

3.1 The Decomposition Algorithm

The algorithm takes a relational polytope complex, that is, a sequence $C_n, \ldots, C_0$ of element sets and a sequence $R_{n-1}, \ldots R_0$ of sparse boundary matrices and returns a sequence $\mu_n, \ldots, \mu_0$ of functions where each μ_i maps an element e_i in C_i to a linear combination of oriented i-simplices $\langle v_i, \ldots, v_0 \rangle$.

The first step assigns each vertex v in C_0 a unique integer number n_v and so $\mu_0(v) = +\langle n_v \rangle$. This specifies a total ordering of vertices and thus the overall orientation of the complex.

Then each edge in C_1 will get a signed ordered pair of these vertex numbers. This means if edge e has starting point s and terminal point t, then $\mu_1(e) = +\langle n_s, n_t \rangle$, iff $n_s < n_t$ and $\mu_1(e) = -\langle n_t, n_s \rangle$ otherwise. We now show that this is a morphism and demonstrate $\mu_0 \partial_1 = \delta_1 \mu_1$:

By $\partial_1(e) = t - s$ we know $\mu_0 \partial_1(e) = \langle n_t \rangle - \langle n_s \rangle$.

Case 1 [$\mu_1(e) = -\langle n_t, n_s \rangle$]:

Then $n_s < n_t$ does not hold and so by $n_t < n_s$ the simplex $\langle n_t, n_s \rangle$ is a simplex of an oriented complex. But then $\delta_1 \mu_1(e) = -\delta_1 \langle n_t, n_s \rangle = -(\langle n_s \rangle - \langle n_t \rangle) = \langle n_t \rangle - \langle n_s \rangle = \mu_0\, \partial_1(e)$.

Case 2 [$\mu_1(e) = +\langle n_s, n_t \rangle$] is trivial.

So in both cases we have the desired equality.

Now the algorithm iteratively produces μ_{i+1} for the i+1-dimensional elements in C_{i+1} from the previously computed μ_i and the given boundary ∂_i. These steps are a generalization of the above step on the edges.

Let e be an i+1-dimensional element with boundary $\partial_i(e)$. We ignore boundary elements with a zero coefficient. Then $\mu_i \partial_i(e)$ is a linear combination of oriented simplices $\langle n_0, \ldots, n_i \rangle$ of which n_0 is the minimal vertex number. From all these first vertex numbers in $\mu_i \partial_i(e)$ take the minimum m_e and remove each simplex $\langle m_e, \ldots \rangle$ in $\mu_i \partial_i(e)$. Let us denote the remaining linear combination of simplices by π(e). We define the simplex tensor product

$$\langle a, \ldots, b \rangle \otimes \langle c, \ldots, d, e \rangle := \langle a, \ldots, b, c, \ldots, d, e \rangle$$

which merely concatenates tuples and which is left- and right distributive over addition: $(w+x) \otimes (y-z) = w \otimes y + x \otimes y - w \otimes z - x \otimes z$. Then $\mu_{i+1}(\mathrm{e}) := \langle m_e \rangle \otimes \pi(e)$ and will now see, that this is a complex morphism, because μ_i is already a complex morphism.

We have to show the algebraic equality $\delta_i \mu_{i+1}(e) = \mu_i \partial_i(e)$ but we first show that algebraically $\delta_i(\langle m_e \rangle \otimes \mu_i \partial_i(e)) = \mu_i \partial_i(e)$ holds. It is easy to see that

$$\delta(s \otimes \sigma) = \delta(s) \otimes \sigma - (-1)^{\dim s}\, s \otimes \delta(\sigma)$$

holds for every pair s, σ of simplices. But then

$$\begin{array}{lll}
\delta_i(\langle m_e \rangle \otimes \mu_i \partial_i(e)) & = \delta_i(\langle m_e \rangle) \otimes \mu_i \partial_i(e) - \langle m_e \rangle \otimes \delta_{i-1}(\mu_i \partial_i(e)) & \dim \langle m_e \rangle = 0 \\
& = \langle\rangle \otimes \mu_i \partial_i(e) - \langle m_e \rangle \otimes (\delta_{i-1}\, \mu_i)\, \partial_i(e) & \langle\rangle \text{ is neutral for } \otimes \\
& = \quad \mu_i \partial_i(e) - \langle m_e \rangle \otimes (\mu_{i-1}\, \partial_{i-1})\, \partial_i(e) & \mu_i \text{ is a morphism} \\
& = \quad \mu_i \partial_i(e) - \langle m_e \rangle \otimes \mu_{i-1}(0) & \partial_{i-1} \partial_i(e) = 0 \\
& = \quad \mu_i \partial_i(e) - \langle m_e \rangle \otimes 0 & \mu_{i-1} \text{ is linear} \\
& = \quad \mu_i \partial_i(e). &
\end{array}$$

If we compute the boundary of a "degenerate" simplex $\langle m_e,m_e,x,\ldots\rangle$ in $\langle m_e\rangle \otimes \mu_i\, \partial_i(e)$ we get

$$\begin{aligned}\delta\,(\langle m_e,m_e,x,\ldots\rangle) &= \delta\,(\langle m_e,m_e\rangle\otimes\langle x,\ldots\rangle)\\ &= (\langle m_e\rangle - \langle m_e\rangle)\otimes\langle x,\ldots\rangle - \langle m_e,m_e\rangle\otimes\delta(\langle x,\ldots\rangle)\\ &= 0\cdot\langle m_e,x,\ldots\rangle - \langle m_e,m_e\rangle\otimes\delta(\langle x,\ldots\rangle)\end{aligned}$$

and see that the only non-degenerate boundary simplex $\langle m_e,x,\ldots\rangle$ cancels. But—as there are no degenerate simplices in $\mu_i\partial_i(e)$—by the former algebraic equality degenerate boundary elements in $\delta_i(\langle m_e\rangle \otimes \mu_i\partial_i(e))$ also have a zero coefficient. Therefore, by induction, algebraically $\delta_i(\langle m_e\rangle \otimes \mu_i\partial_i(e)) = \delta_i\mu_{i+1}(e)$ holds because μ_0, too, does not have degenerate simplices.

3.2 A Small Decomposition Example

We take our 2-d example from Image 1 and define μ_0 by arbitrarily enumerating vertices:

$$\mu_0 : f \mapsto \langle 1\rangle,\ \ b \mapsto \langle 2\rangle,\ \ e \mapsto \langle 3\rangle,\ \ c \mapsto \langle 4\rangle,\ \ a \mapsto \langle 5\rangle,\ \ d \mapsto \langle 6\rangle.$$

Now edge be has boundary $\partial_0(be) = e - b$. Then $\mu_0\partial_0(be) = \langle 3\rangle - \langle 2\rangle$ and vertex 2 is our minimal m_e. So $\mu_1(be) := \langle 2\rangle \otimes \langle 3\rangle = +\langle 2,3\rangle$. Edge cf , on the other hand, has boundary $\partial_0(cf) = f - c$. Then $\mu_0\partial_0(cf) = \langle 1\rangle - \langle 4\rangle$ and so $m_e = 1$. Therefore $\mu_1(cf) = -\langle 1,4\rangle$. Note both the changed orientation and the negative sign. The images of the other edges are:

$$\mu_1 : ab \mapsto -\langle 2,5\rangle,\ \ ad \mapsto +\langle 56\rangle,\ \ bc \mapsto -\langle 2,4\rangle,\ \ de \mapsto -\langle 3,6\rangle,\ \ ef \mapsto -\langle 1,3\rangle$$

We now decompose face A with $\partial_1(A) = -ab + ad - be + de$ to get $\mu_2(A)$: First compute

$$\begin{aligned}\mu_1\partial_1(A) &= -\mu_1(ab) + \mu_1(ad) - \mu_1(be) + \mu_1(de)\\ &= +\langle 2,5\rangle + \langle 5,6\rangle + \langle 2,3\rangle - \langle 3,6\rangle.\end{aligned}$$

Here the minimal m_e is 2 and so $\mu_2(A) = \langle 2\rangle \otimes (\langle 5,6\rangle - \langle 3,6\rangle) = \langle 2,5,6\rangle - \langle 2,3,6\rangle$. The simplicial boundary of $\mu_2(A)$, indeed shows that μ_2 is a morphism:

$$\delta_1\mu_2(A) = \delta_1\langle 2,5,6\rangle - \delta_1\langle 2,3,6\rangle = (\langle 5,6\rangle - \langle 2,6\rangle + \langle 2,5\rangle) - (\langle 3,6\rangle - \langle 2,6\rangle + \langle 2,3\rangle) = \mu_1\partial_1(A)$$

Labelling Image 1 with the labels from μ_0 will show that the positive triangle $\langle 2,5,6\rangle$ makes a counter-clock-wise turn whereas triangle $\langle 2,3,6\rangle$ runs clock-wise. When the complex is of higher dimension n we simply use μ_2 to compute μ_3 etc. until we reach μ_n.

4 Simplex Intersection

We now briefly sketch our geometric n-d intersection algorithm for pair-wise intersection of the achieved triangles. Note that until now we ignored geometry.

An intersection point p of two simplices $S=\langle a,b,c,\ldots\rangle$ and $Z=\langle x,y,\ldots\rangle$ can be found by the active-set-method (Nocedal et. al. 2006) as described in (Paul 2009). It also finds all pairs of boundary simplices that intersect in one point. Then these points' convex closure is the simplex intersection. We start with all vertex pairs in $S_0\times Z_0$ (e.g. $(\langle a\rangle, \langle x\rangle)$). If they are equal they intersect and we are done. Otherwise we determine by gradient computation the vertex to be added to the equation system—the so-called KKT-system (Nocedal et. al. 2006). If e.g. $(\langle a\rangle, \langle x\rangle)$ are disjoint and edge $\langle a,c\rangle$ is closer to $\langle x\rangle$ then the gradient of c is negative and (‹*a*›, ‹*x*›) will be modified to (‹*a*,*c*›, ‹*x*›). Vertex *c* is then said to be "made inactive". The solution of the corresponding KKT-system tells us, if ‹*a*,*c*› and ‹*x*› intersect, and, otherwise, if another vertex could further minimize the distance. The following example shows two intersecting triangles, their intersection, and the intersection vertices.

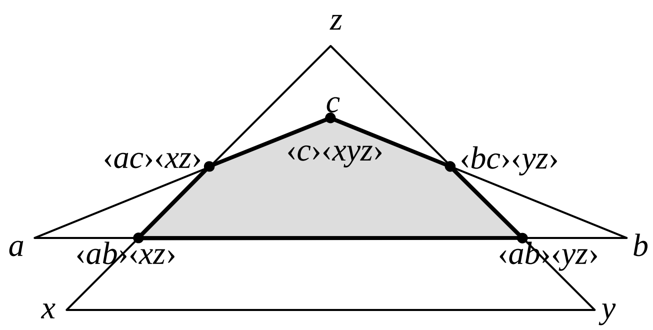

Image 2: The intersection of two simplices ‹*abc*› and ‹*xyz*›. The intersection vertices are labelled with the corresponding intersecting (boundary) simplexes.

All unions of such vertex simplex-pairs produce the simplex-pairs of the higher-dimensional intersections. For example, the union of the points ‹*ac*›‹*xz*› and ‹*c*›‹*xyz*› is ‹*ac*›‹*xyz*›—their connecting edge and also the intersection of the simplices ‹*ac*› and ‹*xyz*›.

Now the tensor product boundary $S \otimes Z$ restricted to the simplices that do intersect gives a valid complex boundary (Lefschetz 1926). The simplicial boundary of, say, edge ‹*abc*›‹*yz*› is

$$\delta(\langle abc\rangle\langle yz\rangle) = \delta(\langle abc\rangle \otimes \langle yz\rangle) = + \langle bc\rangle\langle yz\rangle - \langle ac\rangle\langle yz\rangle + \langle ab\rangle\langle yz\rangle - \langle abc\rangle\langle z\rangle + \langle abc\rangle\langle y\rangle.$$

Removing the vertices that are not on Image 2 gives $\partial(\langle abc\rangle\langle yz\rangle) = +\langle bc\rangle\langle yz\rangle + \langle ab\rangle\langle yz\rangle$, hence a strange edge that has two ending vertices and no starting vertex.

5 Summing Signed Simplex Intersections

The intersections of signed simplices can be summed up to a complex by taking the product of their signs as linear coefficients. But this sum has undesirable features, like the above strange edges. We will only sketch the main correcting steps that finally produce the intersection complex:

1. *Identify and remove duplicate intersection elements*: Intersecting non-convex objects may produce different "non-intersections" of a vertex with a "cavity" which occur as an even number (say, two) of intersections: One with the convex closure and another with a negative volume that carves out a cavity. These can be found combinatorially without using geometry.

2. *Separate connected components*: When an edge intersects a non-convex object the resulting intersection may become a sequence of "dashes". These and also the higher-dimensional connected components are to be identified and separated.

3. *Select an orientation*: The unoriented objects, like edges with two ending points, are oriented. It is still open if a unique orientation can be "naturally" computed from the simplex intersections, or if there must always be a non-deterministic choice from more possibilities.

This then gives a polytope intersection complex $A \cap B$ from two polytope complexes A and B.

6 Conclusion and Outlook

An algorithm was presented which decomposes an polytope complex of arbitrary dimension, complexity and shape into positive and negative *n*-d-simplices and another algorithm which intersects and recombines them to an intersection complex. The algorithm is efficient when the dimension upper bound is fixed. On the other hand, we suppose that with unlimited (dynamic) dimension no polytope intersection algorithm can escape the famous "curse of dimensionality". Currently we work on the open questions regarding the final correction steps to get the resulting complex, in particular, how to compute the orientation of intersection elements. Note that our simplicial tensor product resembles the famous formula of (Eilenberg and Zilber 1953)

$$\partial(a \otimes b) := \partial(a) \otimes b + (-1)^{\dim a}\, a \otimes \partial(b)$$

for tensor product complexes which only differs by a "+" instead of a "–". These alternative boundaries for a tensor product complex—the complex made of the Cartesian product of the elements—provide different orientations of its elements. Keeping in mind that Cartesian product is an important query operator for databases one should see the fundamental importance of resolving this ambiguity.

7 Acknowledgements

This work is funded by the Deutsche Forschungsgemeinschaft (DGF) with research grants BR2128/12-2 and FOR-1546. The authors also acknowledge valuable discussions with Dr. Patrick Erik Bradley.

8 Literature

Bulbul, R. and Frank, A. U. (2009). AHD: The alternate hierarchical decomposition of nonconvex polytopes, *17th International Conference on Geoinformatics*, 1-6.

Chazelle, B. (1984). Convex Partitions of Polyhedra: A Lower Bound and Worst-Case Optimal Algorithm. In: *SIAM Journal on Computing*, 13(3), 488-507.

Cohen J. and Hickey T. (1979). Two Algorithms for Determining Volumes of Convex Polyhedra. *Journal of the ACM*, 26, (3) , 401-414.

Eilenberg, S. and Zilber, J. A. (1953). On Products of Complexes. In: *American Journal of Mathematics*, 75 (1), 200–204

Hatcher, A. (2002). *Algebraic Topology*. Cambridge: Cambridge University Press.

Lefschetz, S. (1926), Intersections and Transformations of Complexes and Manifolds. In: *Transactions of the American Mathematical Society*, 28(1), 1-49.

Nocedal, J and Wright, S. J. (2006). *Numerical Optimization (2. ed.)*. New York: Springer.

Paul, N. (2009). Nicht-konvexe Abstandsbestimmung und Quadratische Programmierung. In: *Forum Bauinformatik 2009*, Universität Karlsruhe (TH).