

IA*: An Adjacency-Based Representation for Non-Manifold Simplicial Shapes in Arbitrary Dimensions

David Canino^a, Leila De Floriani^a, Kenneth Weiss^b

^aDepartment of Computer Science, University of Genova, Italy

^bDepartment of Computer Science, University of Maryland, MD, USA

Abstract

We propose a compact, dimension-independent data structure for *manifold*, *non-manifold* and *non-regular* simplicial complexes, that we call the *Generalized Indexed Data structure with Adjacencies (IA* data structure)*. It encodes only top simplices, i.e., the ones that are not on the boundary of any other simplex, plus a suitable subset of the adjacency relations. We describe the IA* data structure in arbitrary dimensions, and compare the storage requirements of its two-dimensional and three-dimensional instances with both dimension-specific and dimension-independent representations. We show that the IA* data structure is more cost effective than other dimension-independent representations and is even slightly more compact than the existing dimension-specific ones. We present efficient algorithms for navigating a simplicial complex described as an IA* data structure. This shows that the IA* data structure allows retrieving all topological relations of a given simplex by considering only its local neighborhood and thus it is a more efficient alternative to incidence-based representations when information does not need to be encoded for boundary simplices.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.6]: Methodology and Techniques—Graphics data structures and data types

Keywords: Simplicial complexes; non-manifold data structures; topological data structures

1. Introduction

Simplicial complexes are commonly used to describe non-manifold objects in several applications, including finite element analysis, solid modeling, animation, terrain modeling and visualization of scalar and vector fields. Informally, a *manifold* (with boundary) is a compact and connected subset of the Euclidean space for which the neighborhood of each of its points is homeomorphic to an open ball, or to an open half-ball. Objects that do not fulfill this property at one or more points are called *non-manifold*, while they are called *non-regular* if they also contain parts of different dimensions.

Existing modeling tools are generally designed to handle only shapes with a manifold domain [6], since they are topologically simpler. However, non-manifold and non-regular objects arise in several applications. For example, in the *idealization* process for preparing

an object for finite element simulations, regions representing beam or plate behavior are substituted with one-dimensional lines or two-dimensional surfaces, respectively, resulting in objects that contain non-manifold singularities and parts of different dimensions.

In the literature, both dimension-specific and dimension-independent data structures have been developed for cell and simplicial complexes. The former typically exploit properties of the embedding domain to reduce storage requirements, while the latter are independent of the embedding domain. Dimension independent *incidence-based* representations for cell [10] and simplicial [4, 7] complexes encode all the cells of the complex as well as a subset of the incidence relations among the entities, and thus they tend to be quite verbose. In the case of simplicial complexes, *adjacency-based* data structures for manifold objects, and for 2D and 3D non-manifold objects have been shown to be more compact and efficient, since they encode only the *top simplices*, i.e. those simplices not on the boundary of other simplices, as well as the adjacency relations among such simplices [6]. They have, therefore, been used in appli-

Email addresses: canino@disi.unige.it (David Canino), deflo@disi.unige.it (Leila De Floriani), kweiss@cs.umd.edu (Kenneth Weiss)

cations which do not require an explicit encoding of all the simplices.

Here, we propose a dimension-independent data structure for simplicial complexes, that we call the *Generalized Indexed Data Structure with Adjacencies (IA* data structure)*, which generalizes the well known Indexed data structure with Adjacencies (IA data structure) [18]. We show that the IA* data structure is able to retrieve all the relations among the simplices of a simplicial complex very efficiently, it is more compact than all the incidence-based representations and is even slightly more compact than the dimension-specific ones [8, 5]. This is interesting since the latter are able to exploit dimension-specific properties of their embedding space, such as the linear ordering of triangles and tetrahedra around an edge in 3D, to reduce their storage requirements. Furthermore, the IA* data structure is highly *scalable* to the manifold case. This means that, although it can handle arbitrary simplicial complexes, it has negligible overhead, compared to the IA data structure, in manifold regions of the dataset. This is important since, in a typical modeling scenario, the non-manifold irregularities in the shape arise in relatively few localized regions of the shape.

The remainder of this paper is organized as follows. In Section 2, we summarize background notions about simplicial complexes and topological relations, while in Section 3, we review related work. In Section 4, we describe the IA* data structure, its 2D and 3D instances and their storage costs. We present navigation algorithms for efficiently retrieving topological relations among the simplices of a complex in Section 5. In Section 6, we compare the IA* data structure with dimension-independent and dimension-specific data structures for simplicial complexes. Finally, in Section 7, we draw some concluding remarks and discuss our current implementation of the IA* data structure as well as future developments.

2. Background notions

A Euclidean simplex σ of dimension d is the convex hull of $d+1$ linearly independent points in the n -dimensional Euclidean space E^n , where $d \leq n$ is called the *dimension* of σ and is denoted $\dim(\sigma)$. We refer to a *Euclidean d -simplex* as a *d -simplex*: a 0-simplex is a *vertex*, a 1-simplex is an *edge*, a 2-simplex is a *triangle*, and a 3-simplex is a *tetrahedron*. Any Euclidean k -simplex σ' , with $k < d$, generated by a set of vertices $V_{\sigma'} \subseteq V_{\sigma}$ of cardinality $k+1 \leq d$, is called a *k -face* of σ . Where no ambiguity arises, the dimension of σ' can be omitted and σ' is simply called a *face* of σ .

A finite collection Σ of Euclidean simplices forms a *Euclidean simplicial complex* when (i) for each simplex $\sigma \in \Sigma$, all the faces of σ belong to Σ , and (ii) for each pair of simplices σ and σ' , either $\sigma \cap \sigma' = \emptyset$ or $\sigma \cap \sigma'$ is a face of both σ and σ' . The maximal dimension d of the simplices in Σ is called the *order*, or the *dimension* of complex Σ . The *domain*, or *carrier*, of a d -dimensional Euclidean simplicial complex Σ embedded in E^n , with $d \leq n$, is the subset of E^n defined by the union, as point sets, of all the simplices in Σ .

The (*combinatorial*) *boundary* of a simplex σ is the set of all the faces of σ in Σ , excluding σ . The *star* of a simplex σ , denoted $St(\sigma)$ is the set of simplices in Σ that have σ as a face. Any simplex σ such that $St(\sigma)$ contains only σ is called a *top* simplex. The *link* of a simplex σ , denoted $Lk(\sigma)$, is the set of all the faces of the simplices in $St(\sigma)$ which are not incident in σ .

Two simplices are *incident* to each other if one of them is a face of the other, while they are *k -adjacent* if they share a k -face: in particular, two vertices are called *adjacent* if they are both incident at a common edge. An *h -path* is a sequence of $(h+1)$ -simplices $(\sigma_i)_{i=0}^k$ in a simplicial complex Σ such that two consecutive simplices σ_{i-1} and σ_i in the sequence are h -adjacent. Two simplices σ and σ^* are *h -connected* when there exists an h -path $(\sigma_i)_{i=0}^k$ such that σ is a face of σ_0 and σ^* is a face of σ_k . A subset Σ^* of a complex Σ is called *h -connected* if and only if any two simplices of Σ^* are h -connected. Any maximal h -connected sub-complex of a complex Σ is called an *h -connected component* of Σ .

A d -complex Σ in which all top simplices are d -simplices is called *regular*. A regular $(d-1)$ -connected d -complex in which the star of all $(d-1)$ -simplices consists of one or two simplices is called a (*combinatorial*) *pseudo-manifold*. A *manifold* simplicial complex is a pseudo-manifold where the link of any k -simplex, with $k < d-1$, is homeomorphic to the $(d-k)$ -sphere.

Let us consider a simplicial d -complex Σ and a p -simplex $\sigma \in \Sigma$, with $0 \leq p \leq d$. The *topological relations* are defined over Σ in terms of the incidence and adjacency among its simplices:

Boundary relation $R_{p,q}(\sigma)$, with $0 \leq q < p$, consists of the set of q -simplices that are faces of σ .

Co-boundary relation $R_{p,q}(\sigma)$, with $p < q \leq d$, consists of the set of q -simplices incident in σ .

Adjacency relation $R_{p,p}(\sigma)$ consists of the set of p -simplices in Σ that are $(p-1)$ -adjacent to σ (when $p > 0$), or the set of 0-simplices that are adjacent to σ through an edge (when $p=0$).

We call *constant* any relation which involves a constant number of entities, while relations which involve a variable number of entities are called *variable*. In general,

the co-boundary and adjacency relations in a simplicial complex are variable, while the boundary relations are constant. A constant relation should be retrieved by a data structure representing Σ in constant time, while variable relations for a simplex σ should be retrieved by examining a local neighborhood of σ . If the retrieval of a relation requires examining all the simplices of a specific dimension, then the data structure does not support an efficient retrieval of that relation.

3. Related work

Dimension-independent data structures, such as the *cell-tuple* [1] and the *n-G-map* [14], have been proposed for encoding d -dimensional manifold cell complexes. The *Incidence Graph (IG)* [10] represents a general cell complex by encoding all the cells of the complex as well as boundary relations $R_{p,p-1}$ and co-boundary relations $R_{p-1,p}$. It provides a complete, but verbose, description of the complex. Simplified versions of the incidence graph specific for simplicial complexes have been proposed in [4, 7]. Both data structures encode all the simplices in a simplicial complex and subsets of the relations (which we denote as $R_{p,q}^*$) represented in the IG. The *Indexed data structure with Adjacencies (IA)* is a dimension-independent data structure for manifold simplicial d -complexes embedded in the n -dimensional Euclidean space [18]. It encodes only the d -simplices in the complex together with boundary relation $R_{d,0}$ and adjacency relation $R_{d,d}$. Encoding an additional top d -simplex in the star of each vertex (i.e. $R_{0,d}^*$) enables the efficient retrieval of all topological relations.

Several dimension-specific data structures have been developed for manifold two-dimensional cell complexes [6], whereas some are specific for triangle meshes, e.g. the *corner table* [19]. Data structures have been proposed for modeling non-manifold and non-regular objects whose boundary is discretized as a cell complex. The *Partial Entity (PE)* data structure [13] is the most *scalable* to the manifold case, but it is still verbose when applied to simplicial 2-complexes [6]. Dimension-specific data structures have been also proposed for encoding simplicial 2-complexes [2, 8, 17]. The *Loop Edge-use (LE)* [17] and the *Directed Edge (DE)* [2] data structures are for regular simplicial complexes in which non-manifold singularities occur only at edges. The former is a specialization of the PE data structure, while the latter is an extension of the half-edge data structure [16] to non-manifold shapes. The *Triangle-Segment (TS)* data structure [8] is an adjacency-based representation for simplicial 2-complexes embedded in 3D space. It encodes

all the triangles and vertices of the complex, but only the edges which are top simplices. A comparison among such data structures, presented in [6], shows that the TS data structure requires about half of the space compared to the other two. The *eXtended Map (X-Map)* [3] data structure generalizes the combinatorial maps described in [1, 14], to handle non-manifold and non-regular simplicial 2-complexes embedded in E^3 , and can be more expensive than the IG and its variants.

Only a few representations have been proposed in the literature for three-dimensional manifold complexes. The *Facet-Edge (FE)* [9] and the *Handle-Face (HF)* [15] data structures both describe three-dimensional cells implicitly by encoding the manifold complexes that form their boundary. In [12], a scalable data structure for manifold tetrahedral complexes has been proposed, which extends the corner table [19] to the 3D case. An efficient extension of the corner table to tetrahedral meshes has been recently proposed in [11]. The *Non-Manifold Indexed data structure with Adjacencies (NMIA)* [5] is a dimension-specific extension of the IA data structure for non-manifold tetrahedral meshes embedded in 3D space. It encodes all tetrahedra, but only the top triangles and edges.

4. The IA* data structure

In this section, we introduce a new compact dimension-independent data structure for representing Euclidean simplicial complexes in arbitrary dimensions, that we call the *Generalized Indexed data structure with Adjacencies (IA* data structure)*, which generalizes the IA data structure [18] to non-manifold and non-regular objects discretized as simplicial complexes.

The IA* data structure encodes all the top simplices of a d -dimensional simplicial complex Σ embedded in the n -dimensional Euclidean space (with $d \leq n$) plus the following relations:

- $R_{p,0}(\sigma)$: boundary relation for each top p -simplex σ , where $0 < p \leq d$;
- $R_{p,p}^*(\sigma)$: partial adjacency relation for each top p -simplex σ , where $p > 1$: it consists of all top p -simplices adjacent to σ along a $(p-1)$ -face;
- $R_{p-1,p}^*(\tau)$: partial co-boundary relation for each $(p-1)$ -simplex τ on the boundary of more than two top p -simplices, with $0 < (p-1) < d$: it consists of all top p -simplices incident to τ ;
- $R_{0,1}^*(v)$: partial co-boundary relation for each vertex v : it consists of all top 1-simplices in the star of v ;
- $R_{0,p}^*(v)$: partial co-boundary relation for each vertex v , with $2 \leq p \leq d$: it consists of one arbitrarily selected

top p -simplex for each $(p - 2)$ -connected component Σ' of the link of v , $Lk(v)$, where Σ' is formed by top $(p - 1)$ -simplices in $Lk(v)$;

Relation $R_{p-1,p}^*(\tau)$ encodes the co-boundary relation of the $(p-1)$ -faces τ of a top p -simplex σ such that there are more than two top p -simplices adjacent to σ along τ . This facilitates an efficient encoding for relation $R_{p,p}^*(\sigma)$ along a $(p - 1)$ -face τ as either a single top p -simplex adjacent to σ along τ or as a pointer to the list of all top p -simplices incident in τ (i.e. $R_{p-1,p}^*(\tau)$).

Note that for each d -simplex σ , the encoded $R_{d,d}^*(\sigma)$ is actually the full adjacency relation $R_{d,d}(\sigma)$ since all d -simplices are top simplices. When $d=n$, any simplicial d -complex is a pseudo-manifold in the d -dimensional Euclidean space. In this case, $R_{d,d}(\sigma)$ is a constant relation consisting of one simplex adjacent to σ along each of its $(d - 1)$ -faces. Consequently, $R_{d-1,d}^*(\sigma)$ is empty.

When the complex is regular, all top simplices are d -simplices, so the cardinality of partial co-boundary relation $R_{0,d}^*(v)$ for a vertex v is equal to the number of connected components of the link of v . Since there are no top simplices of dimension less than d , the only boundary relations stored are $R_{d,0}$.

If the domain of Σ is a manifold, the IA^* data structure reduces to the IA data structure. Specifically, $R_{0,p}^*(v)$ is empty for any $0 < p < d$, and $R_{0,d}^*(v)$ contains just one d -simplex since the link of a vertex v consists of a single $(d - 2)$ -connected component. Furthermore, since there are no non-manifold simplices, $R_{p-1,p}^*$ is always empty.

4.1. The IA^* data structure in 2D and 3D

We consider here two instances of the IA^* data structure for simplicial 2- and 3-complexes in 3-dimensional Euclidean space E^3 , that we call, for brevity, the $IA^*(2D)$ and $IA^*(3D)$ data structures.

The $IA^*(2D)$ data structure encodes the vertices as well as all triangles (top 2-simplices) and all *wire edges* (top 1-simplices), i.e., edges not on the boundary of any triangle, plus the following relations:

- $R_{1,0}(\sigma)$: boundary relation for each wire edge;
- $R_{2,0}(\sigma)$: boundary relation for each triangle;
- $R_{2,2}(\sigma)$: adjacency relation for each triangle σ : it consists of all triangles adjacent to σ along an edge;
- $R_{1,2}(e)$: co-boundary relation for non-manifold edge e , consisting of all triangles incident to e ;
- $R_{0,1}^*(v)$: partial co-boundary relation for each vertex v : it consists of all wire edges in the star of v ;
- $R_{0,2}^*(v)$: partial co-boundary relation for each vertex v : it consists of one triangle from each 0-connected component of $Lk(v)$ formed by at least one edge.

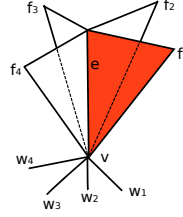


Figure 1: a simplicial 2-complex embedded in E^3 . Partial adjacency relations $R_{2,2}^*$ for a triangle f_1 (red) along non-manifold edge e points to $R_{1,2}^*(e) = \{f_1, f_2, f_3, f_4\}$, which is encoded only once for all triangles incident to e . Encoded co-boundary relations for vertex v are: $R_{0,1}^*(v) = \{w_1, w_2, w_3, w_4\}$ and $R_{0,2}^*(v) = \{f_1\}$, where f_1 is arbitrarily selected.

In a simplicial 2-complex embedded in E^3 , an edge e is non-manifold if more than two triangles are incident to e . Consequently, we can encode $R_{2,2}$ along a non-manifold edge e as a pointer to the list of all triangles incident to e , i.e. $R_{p-1,p}^*(e)$. Figure 1 illustrates the relations encoded by the $IA^*(2D)$ data structure for a small simplicial 2-complex in E^3 .

The $IA^*(3D)$ data structure encodes the vertices as well as the tetrahedra (top 3-simplices), the *dangling faces* (top 2-simplices) and *wire edges* (top 1-simplices) plus the following relations:

- $R_{k,0}(\sigma)$: boundary relation for each top k -simplex σ ;
- $R_{3,3}(\sigma)$: adjacency relation for each tetrahedron σ : it consists of the four tetrahedra adjacent to σ along the triangles bounding σ ;
- $R_{1,2}^*(e)$: partial co-boundary relation for each non-manifold edge e on the boundary of a top triangle: it consists of all top triangles incident to e ;
- $R_{2,2}^*(\sigma)$: partial adjacency relation for each dangling face σ : it consists of all top 2-simplices adjacent to σ along their edges;
- $R_{0,1}^*(v)$: partial co-boundary relation for each vertex v : it consists of all top 1-simplices in the star of v ;
- $R_{0,2}^*(v)$: partial co-boundary relation for each vertex v : it consists of one one triangle for each 0-connected component in $Lk(v)$ that consists only of top 1-simplices in $Lk(v)$;
- $R_{0,3}^*(v)$: partial co-boundary relation for each vertex v : it consists of one tetrahedron for each 1-connected component of $Lk(v)$ formed by triangles in $Lk(v)$;

In a simplicial 3-complex embedded in E^3 , an edge e is non manifold when its link consists of more than one connected component. Thus, we can encode relation $R_{2,2}$ along a non-manifold edge e as a pointer to all dangling triangles (top 2-simplices) incident in e , i.e. $R_{1,2}^*(e)$. Figure 2 illustrates relations encoded by the $IA^*(3D)$ data structure for a simplicial 3-complex in E^3 .

4.2. Implementation and storage costs

In our implementation of the IA^* data structure, each simplex is indexed through an integer value. The vertices are encoded in an array storing the coordinates

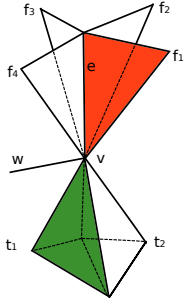


Figure 2: Simplicial 3-complex in E^3 . Adjacency relations among the tetrahedra are $R_{3,3}(t_1) = \{t_2\}$ and $R_{3,3}(t_2) = \{t_1\}$. Partial adjacency relations $R_{2,2}^*$ for the four top triangles f_i along non-manifold edge e points to $R_{1,2}^*(e) = \{f_1, f_2, f_3, f_4\}$. Co-boundary relations for vertex v are $R_{0,1}^*(v) = \{w\}$; $R_{0,2}^*(v) = \{f_1\}$ (red), which is arbitrarily selected from the single 0-connected component of edges in $Lk(v)$; and $R_{0,3}^*(v) = \{t_1\}$ (green), which is arbitrarily selected from the single 1-connected component of triangles in $Lk(v)$.

plus partial co-boundary relations $R_{0,p}^*(v)$, with $p \geq 1$, which are encoded as variable-sized arrays. Boundary relations for top p -simplices (with $p > 1$) are stored in arrays of dimension $p + 1$. For efficiency, partial adjacency relation $R_{p,p}^*(\sigma)$ is encoded as an array with $(p+1)$ elements, each corresponding to a $(p-1)$ -face τ of σ . If τ is manifold, the element of the array corresponding to τ contains the index of the only p -simplex adjacent to σ along τ . Otherwise, it contains a pointer to a variable-size array encoding $R_{p-1,p}^*(\tau)$, which contains the indexes of all top p -simplices incident in τ .

In a simplicial d -complex Σ , we denote with s_i the total number of its i -simplices, and with s_i^t the number of its top i -simplices, with $i = 1, 2, \dots, d$. The IA^* data structure for encoding a d -dimensional simplicial complex Σ embedded in the n -dimensional Euclidean space stores the following number of integer values:

- $(d+1) \cdot s_d$ for boundary relation $R_{d,0}$;
- $(j+1) \cdot s_j^t$ for partial boundary relations $R_{j,0}^*$, for $1 \leq j < d$;
- $(d+1) \cdot s_d$ for adjacency relation $R_{d,d}$;
- $(k+1) \cdot s_k^t$ for partial adjacency relation $R_{k,k}^*$ restricted to the top k -simplices, with $1 < k < d$;
- $\sum_{q=1}^{s_{k+1}} nm_q^k$ for partial co-boundary relation $R_{k,k+1}^*$ restricted to non-manifold k -simplex τ on the boundary of a top $(k+1)$ -simplex;
- $2s_1^t$ for partial co-boundary relation $R_{0,1}^*$;
- $\sum_{j=1}^{s_0} c_q^k$ for partial co-boundary relation $R_{0,k}^*$, $k > 1$;

where $nm_q^k \in \{0, 1, \dots, k+2\}$ denotes the number of non-manifold k -faces in top $(k+1)$ -simplex σ_q and c_q^k denotes the number of $(k-2)$ -connected components of $Lk(v_q)$ consisting only of top $(k-1)$ -simplices in $Lk(v_q)$. Thus, the total cost of the topological relations in the IA^* data structure is equal to:

$$2 \sum_{j=1}^d (j+1)s_j^t + \sum_{k=1}^{d-1} \sum_{q=1}^{s_{k+1}} nm_q^k + \sum_{k=2}^d \sum_{q=1}^{s_0} c_q^k,$$

where each item is an integer value. Substituting C^k for

$\sum_{j=1}^{s_0} c_q^k$ and NM^1 for $\sum_{k=1}^{s_2} nm_k^1$, we can determine the storage cost for $IA^*(2D)$ to be: $6s_2 + 4s_1^t + NM^1 + C^2$, and that of $IA^*(3D)$ to be: $8s_3 + 6s_2^t + 4s_1^t + NM^1 + C^2 + C^3$.

5. Retrieving topological relations

In this section, we show how to retrieve the topological relations from the IA^* data structure encoding a d -dimensional simplicial complex Σ embedded in the n -dimensional Euclidean space. These form the basis for navigating and manipulating the complex.

5.1. Retrieving boundary relations

All boundary relations $R_{p,0}$ are explicitly encoded. Boundary relation $R_{p,q}(\sigma)$ for a top p -simplex σ , with $0 < q < p$, can be simply retrieved by generating all $(q+1)$ -tuples of vertices forming the q -faces of σ . The time complexity is therefore $O\left(\binom{p+1}{q+1}\right)$.

Boundary relation $R_{p,q}(\sigma)$ for a non-top p -simplex σ , with $q < p$, can be retrieved only when σ is specified as a p -face of a top h -simplex τ , with $h > p$. In this case, the q -simplices which are faces of σ can be simply retrieved as above.

5.2. Retrieving co-boundary relations

We consider first co-boundary relations based on the vertices, which allow us to retrieve the simplices incident in any given vertex v . Relation $R_{0,d}(v)$, i.e. all d -simplices incident in v , is retrieved by starting from the d -simplices in $R_{0,d}^*(v)$ and navigating the complex by using the explicitly stored $R_{d,d}$ relation. $R_{0,d}^*(v)$ gives us one d -simplex for each $(d-1)$ -connected cluster of d -simplices incident in vertex v . Specifically, given the d -simplex σ encoded in $R_{0,d}^*(v)$, we initialize the cluster with σ and retrieve all d -simplices in the same cluster as σ by a breadth-first traversal using relation $R_{d,d}$.

Relation $R_{0,d-1}(v)$, i.e., the collection of all $(d-1)$ -simplices incident in v , is retrieved in two stages. First, we retrieve $R_{0,d}(v)$ as above, and, for each d -simplex σ in $R_{0,d}(v)$, we extract its $(d-1)$ -faces through boundary relation $R_{d,d-1}(\sigma)$. We then apply a breadth-first traversal of the elements in $R_{0,d-1}^*(v)$ using a similar algorithm as above for $R_{0,d}^*(v)$. Thus, the retrieval of any $R_{0,p}(v)$ can be performed by retrieving relations $R_{0,j}(v)$ and $R_{j,p}(\sigma)$, with $j > p$, for every j -simplex σ in $R_{0,j}(v)$, for $j = d, d-1, \dots, p$. The time complexity can be easily shown to be linear in the total number of top simplices of dimension $p, p+1, \dots, d$ incident in v .

Relation $R_{p,q}(\sigma)$, with $p < q$, for a p -simplex σ is specified by describing σ as a $(p+1)$ -tuple $[v_0, v_1, \dots, v_p]$ of vertices. We then retrieve $R_{0,q}(v_0)$ and we select all

the q -simplices τ in $R_{0,q}(v_0)$ such that all the other vertices (v_1, \dots, v_p) in the $(p+1)$ -tuple are in $R_{q,0}(\tau)$ as well. For instance, for retrieving $R_{1,3}(e)$ of an edge $e = (v_0, v_1)$, we need to retrieve all tetrahedra incident in v_0 through $R_{0,3}(v_0)$ and then we select only those tetrahedra containing vertex v_1 in their boundary. The time complexity of the algorithm for retrieving $R_{p,q}(\sigma)$ is dominated by the complexity of retrieving relation $R_{0,q}(v_0)$, which is linear in the number of top simplices incident in v_0 of dimension from q to d .

5.3. Retrieving adjacency relations

Adjacency relation $R_{d,d}(\sigma)$ is stored in the IA* data structure for each d -simplex σ in Σ . Partial relation $R_{p,p}^*(\tau)$, $p > 1$, is stored for every top p -simplex τ and is restricted to the top p -simplices $(p-1)$ -adjacent to τ .

Adjacency relation $R_{p,p}(\sigma)$, $p > 0$, for a top p -simplex σ is extracted by considering $R_{p,p}^*(\sigma)$, which gives us top p -simplices $(p-1)$ -adjacent to σ . The other p -simplices, which are not top simplices, are retrieved by combining boundary relation $R_{p,p-1}$ and co-boundary relation $R_{p-1,p}$. Boundary relation $R_{p,p-1}(\sigma)$ consists of $(p+1)$ p -tuples of vertices formed by σ 's vertices. For each of them, we retrieve $R_{p-1,p}$ as above. Note that $R_{1,1}^*$ does not need to be encoded in the IA* data structure.

If σ is not a top simplex, then σ must be specified as a p -face of a top h -simplex τ , with $h > p$. In this case, $R_{p,p}^*(\sigma)$ is empty and thus all p -simplices in $R_{p,p}(\sigma)$ need to be retrieved by combining boundary relation $R_{p,p-1}(\sigma)$ with co-boundary relation $R_{p-1,p}$.

Adjacency relation $R_{0,0}(v)$ for a vertex v can be retrieved in exactly the same way as $R_{0,1}(v)$, since the edges in $R_{0,1}(v)$ are expressed as pairs of vertices.

The time complexity for retrieving adjacency relation $R_{p,p}(\sigma)$, $p > 0$, is dominated by the time required for retrieving $R_{p-1,p}(\sigma)$, which is linear in the number of top simplices incident in one of the vertices of σ .

6. Analysis and comparisons

Recall that there are dimension-independent and dimension-specific data structures for simplicial complexes. Dimension-independent data structures for simplicial complexes are incidence-based, namely the Incidence Graph (IG) [10] and its specialized instances, the *Simplified Incidence Graph (SIG)* [4] and the *Incidence Simplicial (IS)* [7] data structures. In the following, we will restrict our dimension-independent comparisons to the IS which has been shown to be the most compact [7]. Regarding dimension-specific data structures, it has been shown that the TS and the NMIA

data structures are respectively the most compact 2D and 3D representations for simplicial complexes and, thus, we will restrict our dimension-specific comparisons of the IA* data structure to these data structures. For brevity, we refer the reader to [7, 8, 5] for details on these data structures.

Incidence-based data structures. Given a d -simplicial complex Σ embedded in E^n (with $d \leq n$), the IS data structure [7] stores *all* simplices of Σ , the full boundary relations $R_{p,p-1}$ and the partial co-boundary relations $R_{p,p+1}^*$, i.e. one $(p+1)$ -simplex for each connected component in the link of each p -simplex. Let h_q^p be the number of connected components in the link of a p -simplex σ_q , then the IS requires $\sum_{0 < p \leq d} (p+1) \cdot s_p + \sum_{p=0}^{d-1} \sum_{q=1}^{s_p} h_q^p$ integers. In the link of a simplex σ , we can identify connected components related to top simplices and components related to non top-simplices, indicated as $K^i = \sum_{q=1}^{s_p} h_q^i$. Thus, the IS requires $6s_2 + 2s_1 + 2s_1^t + K^0$ integers in 2D, and $8s_3 + 3s_2 + 2s_1 + 3s_2^t + 2s_1^t + K^0 + K^1$ integers in 3D.

Adjacency-based structures. The primary difference between the dimension-specific TS and NMIA data structures from the IA*(2D) and IA*(3D) data structures is that the former exploit their embedding in E^3 in their encodings of the topological relations. Specifically, whereas the IA*(2D) encodes non-manifold edges of triangles in $R_{2,2}^*$ using a pointer to the shared information in $R_{1,2}^*$, the TS data structure encodes $R_{2,2}^*$ as a doubly-linked list around the non-manifold edges of its triangles. That is, each triangle along non-manifold edge e stores a pointer to the previous triangle and the next triangle surrounding e in a clockwise orientation. Over the entire data structure, $R_{2,2}^*$ requires $3s_2 + 2NM^1$ integers, giving the TS data structure a total cost of: $6s_2 + 4s_1^t + C^2 + 2NM^1$ integers.

The NMIA encodes its boundary relations similarly to the IA*(3D), its vertex co-boundary relations similarly to the IS data structure, i.e. one element per connected component, and orients the simplices in the link of non-manifold edges in clockwise order similarly to the TS data structure. It encodes for each *cluster* (a dangling face or a fan of tetrahedra) incident to non-manifold e , the preceding and following cluster surrounding e . Letting $NM^{1,3}$ be the total number of such fans of tetrahedra and $NM^{1,2}$ be the total number of dangling faces in the mesh (which is equivalent to NM^1 for the IA*(3D)), the cost of the NMIA data structure is $8s_3 + 6s_2^t + 4s_1^t + K^0 + 2NM^1 + 2NM^{1,3}$ integers.

Since the most common applications of these meshes

Table 1: Statistics for representative manifold (top), non-manifold regular (middle) and non-manifold non-regular (bottom) simplicial 2-complexes Σ in E^3 and their storage requirements for the IS(2D), TS and IA*(2D) data structures. Statistics are: the total number of i -simplices (s_i), and the subset that are top simplices (s_i^t) and non-manifold simplices (s_i^n). $C^2 = K^0$ is the sum of connected components in the link of the vertices consisting of edges, while NM^1 is the sum of non-manifold edges on the boundary of triangles in Σ .

Model	s_0	s_1	s_2	s_1^t	s_0^n	s_1^n	C^2	NM^1	IS(2D)	TS	IA*2D
Football	1.2 K	3.7 K	2.5 K	–	–	–	1.2 K	–	23.4 K	16.0 K	16.0 K
2D torus	10.4 K	30.7 K	20.5 K	–	–	–	10.4 K	–	195 K	133 K	133 K
Pinched torus	1.8 K	5.5 K	3.7 K	–	1	–	.3 K	–	35 K	24 K	24 K
Armchair	5.3 K	15.9 K	10.6 K	–	57	56	5.3 K	.2 K	101 K	69.4 K	69.2 K
Robot	3.2 K	10.0 K	6.7 K	–	508	480	3.2 K	1.4 K	63.5 K	46.4 K	45 K
Balance	4.1 K	12.0 K	8.0 K	34	8	–	4 K	–	75.9 K	51.9 K	51.9 K
Chandelier	9.2 K	27.6 K	18.3 K	136	352	264	9.2 K	.8 K	175 K	121 K	120 K
Tower-wire	8.3 K	24.6 K	15.9 K	896	1.3 K	795	7.9 K	2.4 K	154 K	112 K	109 K

Table 2: Statistics for representative manifold (top), non-manifold regular (middle) and non-manifold non-regular (bottom) simplicial 3-complexes Σ in E^3 and their storage requirements for the IS(3D), NMIA and IA*(3D) data structures. s_i are the total number of i -simplices, while s_i^t and s_i^n are the subset that are top simplices and non-manifold simplices. C^2 and C^3 are the sum of 0-connected and 1-connected components in the link of the vertices that consist of edges and triangles, respectively. K^0 and K^1 are the sum of connected components in the links of the vertices and edges of the mesh, respectively. $NM^1 = NM^{1,2}$ and $NM^{1,3}$ are the sums of non-manifold edges on the boundary of top triangles and tetrahedra in Σ .

Model	s_0	s_1	s_2	s_3	s_1^t	s_2^t	s_0^n	s_1^n	C^2	C^3	K^0	K^1	NM^1	$NM^{1,3}$	IS(3D)	NMIA	IA*(3D)
Rings	1.2 K	6.4 K	9.2 K	4.0 K	–	–	–	–	–	1.2 K	1.2 K	6.4 K	–	–	80.2 K	33.2 K	33.2 K
Gargoyle	2.7 K	14.7 K	22.0 K	10.0 K	–	–	–	–	–	2.7 K	2.7 K	14.7 K	–	–	193 K	82.7 K	82.7 K
Pinched torus	.3 K	1.3 K	1.9 K	853	–	–	10	9	–	262	253	1.3 K	–	18	16.7 K	7.1 K	7.1 K
Wheel	.4 K	2.1 K	2.7 K	1.1 K	96	32	112	24	48	402	402	1.9 K	32	80	24.2 K	10.3 L	10.2 K
Balloon	1.1 K	3.9 K	3.6 K	.9 K	64	1632	48	–	818	274	1.1 K	1.4 K	–	–	33.0 K	18.0 K	18.0 K
Flasks	1.3 K	6.3 K	8.5 K	3.5 K	–	469	40	36	242	1.1 K	1.3 K	5.6 K	68	72	74.0 K	32.0 K	31.8 K
Pot	4.7 K	17.9 K	17.0 K	5.7 K	2.9 K	3.9 K	1.1 K	70	2.1 K	1.7 K	3.7 K	9.1 K	75	94	163 K	84.7 K	84.5 K

are for simplicial 2- and 3-complexes embedded in E^3 , we compare the IA*(2D) against the IS(2D) and TS data structures, and the IA*(3D) against the IS(3D) and NMIA data structures. We summarize the storage costs of the different data structures in terms of common parameters, and compare them experimentally against a testbed of 62 simplicial 2- and 3-complexes of varying domains (manifold, regular, non-manifold) and complexity. We highlight a representational subset of the 2D datasets in Table 1 and of the 3D datasets in Table 2.

In 2D, it is evident that the IA*(2D) is more compact than the TS by the term NM^1 . The key difference between the IS(2D) and the IA*(2D) relates to the total number of edges in the mesh compared to the number of wire edges and non-manifold edges, i.e. $2s_1$ in the former vs. $2s_1^t + NM^1$ in the latter. Since the total number of edges in the complex is generally significantly higher than the number of non-manifold and/or wire-edges, we expect the IA*(2D) and TS data structures to be significantly more compact than the IS(2D) data structure, and for the IA*(2D) to be slightly more compact than the TS. Empirically, we found the IS(2D) to require approximately 45% more space than the IA*(2D) in all cases and the TS to require 0-5% more space than the IA*(2D) on a testbed of 43 simplicial 2-complexes that

were manifold (8), non-manifold regular (20) and non-manifold non-regular (15). Table 1 highlights a representative sample of these results.

Similarly, in 3D, the primary difference between the IS(3D) and the IA*(3D) and NMIA data structures relates to the total number of triangles s_2 and edges s_1 compared to the number of top triangles s_2^t and top edges s_1^t , while the IA*(3D) and NMIA differ in terms of the number of top simplices compared to the number of non-manifold edges, i.e. $NM^1 + C^2 + C^3$ vs. $2NM^1 + 2NM^{1,3} + K^0$. To better understand the consequences of these different encodings, we analyze properties of 19 simplicial 3-manifolds in E^3 that are manifold (6), regular non-manifolds (3) and non-manifold non-regular (10) complexes, and present representative samples in Table 2. We observe that $C^2 + C^3$ in the IA*(3D) encoding is approximately equal to K^0 for NMIA, although K^0 can be smaller. Additionally, in the tested datasets, the NM^1 and $NM^{1,3}$ are generally quite small and do not impose a significant overhead.

The IA*(3D) is the most compact data structure in all tested datasets: In the manifold case, the IA*(3D) and NMIA require the same storage, and are both about 1.5 times smaller than the IS(3D) data structure. In all other cases, NMIA requires approximately 3% more

storage than $IA^*(3D)$. In the regular case, the $IS(3D)$ requires about three times the space as $IA^*(3D)$, while it requires 2.2 times the space when dealing with non-regular datasets.

7. Concluding remarks

We have introduced the *Generalized Indexed data structure with Adjacencies* (IA^* data structure), a new dimension-independent data structure for encoding non-manifold and non-regular objects in arbitrary dimensions discretized as d -dimensional simplicial complexes. We feel that the IA^* data structure is the natural generalization of the Indexed data structure with Adjacencies (IA data structure) [18] to the domain of non-manifold and non-regular simplicial shapes, since it is agnostic about its embedding space, and reverts to the IA data structure when presented with manifold simplicial shapes. We compared the IA^* data structure with other dimension-specific and dimension-independent representations, in terms of storage costs and of efficiency in retrieving topological relations. In terms of storage costs the IA^* data structure is more compact than all state of the art dimension-specific and dimension-independent representations, even those that take the embedding space into account. This behavior is even more evident when the number of non-manifold singularities and of non-maximal top simplices is not too high, as is common in most non-manifold shapes.

We have developed a C++ library, the *IA* library* containing a dimension-independent implementation of the IA^* data structure. The current version supports the construction of the data structure as well as the navigation queries described in Section 5. The construction (which we omit for brevity) is performed from a soup of simplices. Currently, we are enhancing the library by developing topological manipulation operators, in particular the *vertex-pair contraction* operator, to enable simplifications and repair on general simplicial complexes. The *IA* library* is part of a set of tools for navigating and manipulating simplicial complexes in arbitrary dimensions, and contains implementations of the IS [7], IG (Incidence graph) [10] and SIG (Simplified Incidence Graph) [4] data structures.

We have performed some experimental comparisons between the IG, the IS, and the IA^* data structures in terms of efficiency of the topological queries. Our initial results indicate that the IA^* data structure is the most efficient when extracting boundary relations, requiring approximately half the time to extract the boundary of a simplex. Extracting the co-boundary of a vertex from the IA^* data structure also requires about half the time

as the IG and the IS data structures. Interestingly, the extraction times for non-vertex simplices in the IA^* are within 5% of those for the IS data structure, despite the need to extract the co-boundary of a vertex incident in that simplex and filter the results in the former case.

References

- [1] Brisson, E., 1989. Representing geometric structures in d dimensions: topology and order. In: Proc. of the 5th ACM Symp. on Computational Geometry. ACM Press, pp. 218–227.
- [2] Campagna, S., Kobbelt, L., Seidel, H.-P., 1998. Directed Edges - a scalable representation for triangle meshes. Journal of Graphics Tools 3 (4), 1–12.
- [3] Cazier, D., Kraemer, P., 2010. X-Maps: an efficient model for non-manifold modeling. In: Proceedings IEEE Shape Modeling International. IEEE Computer Society, pp. 226–230.
- [4] De Floriani, L., Greenfieldboyce, D., Hui, A., 2004. A data structure for non-manifold simplicial d-complexes. In: Proc. Eurographics/ACM Symp. on Geom. Proces. ACM, pp. 83–92.
- [5] De Floriani, L., Hui, A., 2003. A scalable data structure for three-dimensional non-manifold objects. In: Proceedings Eurographics Symposium on Geometry Processing. pp. 72–82.
- [6] De Floriani, L., Hui, A., 2005. Data structures for simplicial complexes: an analysis and a comparison. In: Proc. of the 3rd Eurographics Symp. on Geometry Processing. pp. 119–128.
- [7] De Floriani, L., Hui, A., Panozzo, D., Canino, D., 2010. A dimension-independent data structure for simplicial complexes. In: Proceedings of the IMR. Springer, pp. 403–420.
- [8] De Floriani, L., Magillo, P., Puppo, E., Sobrero, D., 2004. A multi-resolution topological representation for non-manifold meshes. CAD Journal 36 (2), 141–159.
- [9] Dobkin, D., Laszlo, M., 1989. Primitives for the manipulation of three-dimensional subdivisions. Algorithmica 5 (4), 3–32.
- [10] Edelsbrunner, H., 1987. Algorithms in Combinatorial Geometry. Springer.
- [11] Gurung, T., Rossignac, J., 2009. SOT: a compact representation for tetrahedral meshes. In: Proc. of the SIAM/ACM Geometric and Physical Modeling. San Francisco, USA, pp. 79–88.
- [12] Lage, M., Lewiner, T., Lopes, H., Velho, L., 2005. CHF: A scalable topological data structure for tetrahedral meshes. In: Proceedings SIBGRAPI. pp. 349–356.
- [13] Lee, S. H., Lee, K., 2001. Partial-entity structure: a fast and compact non-manifold boundary representation based on partial topological entities. In: Proc. ACM Symp. on Solid Modeling and Applications. pp. 159–170.
- [14] Lienhardt, P., 1991. Topological models for boundary representation: a comparison with n -dimensional generalized maps. CAD Journal 23 (1), 59–82.
- [15] Lopes, H., Tavares, G., May 1997. Structural operators for modeling 3-manifolds. In: Proc. of the 4th ACM Symp. on Solid Modeling and Applications. ACM Press, pp. 10–18.
- [16] Mantyla, M., 1987. An Introduction to Solid Modeling. Computer Science Press.
- [17] McMains, S., 2000. Geometric Algorithms and Data Representation for Solid Freeform Fabrication. Ph.D. thesis, University of California at Berkeley.
- [18] Paoluzzi, A., Bernardini, F., Cattani, C., Ferrucci, V., 1993. Dimension-independent modeling with simplicial complexes. ACM Transactions on Graphics 12 (1), 56–102.
- [19] Rossignac, J., Safonova, A., Szymczak, A., 2001. 3D compression made simple: Edgebreaker with zip & wrap on a corner-table. In: Proc. Shape Modeling International. pp. 278–283.