A Representation for Abstract Simplicial Complexes: An Analysis and a Comparison

Leila De Floriani, Franco Morando, and Enrico Puppo

Department of Computer and Information Sciences, University of Genova
Via Dodecaneso 35, 16146 Genova, Italy

Abstract. Abstract simplicial complexes are used in many application contexts to represent multi-dimensional, possibly non-manifold and non-uniformly dimensional, geometric objects. In this paper we introduce a new general yet compact data structure for representing simplicial complexes, which is based on a decomposition approach that we have presented in our previous work [3]. We compare our data structure with the existing ones and we discuss in which respect it performs better than others.

Keywords: Non-manifold modeling, simplicial complexes,data structures.

1 Introduction

Geometric cell complexes are widely used to represent multi-dimensional geometric objects in many applications. In particular, simplicial complexes have received great attention both from a theoretical and from a practical point of view. In fact, their combinatorial properties make them easier to understand, represent and manipulate than more general cell complexes.

A data structure representing a complex should not only describe its shape unambiguously, but should also support efficient traversal and editing operations [13].

Although most work in the geometric modeling literature has been aimed at representing just three-dimensional *manifold* objects, several authors have pointed out the need of developing more general data structures, which can represent also higher dimensional and/or non-manifold and non-uniformly dimensional objects [8,15,10]. Non-manifold singularities in modeled objects occurs as a side-effect of feature extraction from images, 3D reconstruction or as a byproduct of severe discretization. Sometimes singularities are actually essential when, for instance, we choose to model the semantic content of an image (e.g. [9]) with an object of mixed dimensionality. This generality is usually paid in terms of some overhead in storage costs. On the other hand, most objects encountered in the applications contain a relatively small number of non-manifold *singularities.* Thus, it is important to develop data structures that are not burdened by an excessive overhead, when they are used to represent manifold objects, i.e., they *scale* well with the degree of "non-manifoldness".

I. Nyström et al. (Eds.): DGCI 2003, LNCS 2886, pp. 454–464, 2003.

In this paper, we first review several data structures available from the literature for non-manifold modeling.

Next, we describe a two-level data structure that we call *Non-Manifold Decomposition Data Strucutre* (NMD-DS). The NMD-DS can represent any simplicial complex in any dimension and downscales well to the manifold case. This data structure is based on a scheme for decomposing non-manifold complexes into nearly manifold parts, that we presented in [3]. A complex is decomposed in a unique way into a reduced number of components such that each component is as free as possible from singularities.

Each decomposition component belongs to a well-understood class of complexes, that we called *initial quasi-manifolds.* Such complexes are simple enough to be represented with a data structure, having a cost comparable to those used for representing manifolds. We call this data structure the *Initial Quasi Manifold Data Structure* (IQM-DS). This data structure was introduced in [4] and is detailed here in Section 5.1.

The collection of representations of components constitutes the first level of our data structure. The assembly of all components is represented in the second level, which is designed to support efficient traversal of the complex across different components.

2 Background

Purely geometrical aspects are not relevant in the design of data structures because geometric embedding is always encoded by adding just coordinates to vertices. Therefore, we will address only *abstract* complexes, by focusing on their combinatorial structure and on the topological relations among their cells.

Abstract Simplicial Complexes. Let V be a finite set of elements that we call *vertices*. An *abstract simplicial complex* on V is a subset Ω of the set of (non empty) subsets of V such that: $\{v\} \in \Omega$ for every vertex $v \in V$; and if $\gamma \subset V$ is an element of Ω, then every subset of γ is also an element of Ω. Each element of Ω is called an *abstract simplex*, or just a *simplex.* The *dimension* of a simplex $\gamma \in \Omega$, denoted $\dim(\gamma)$, is the number of vertices in γ minus one. A cell of dimension s is called an s-cell. A complex Ω is called *d-dimensional* or a *d-complex* if $\max_{\gamma \in \Omega}(\dim(\gamma)) = d$. Each d-cell of a d-complex Ω is called a *maximal cell* of Ω.

The set of all cells of dimension *smaller or equal to m* is called the *m-skeleton* of Ω (denoted by Ω^m). The set of all simplices of dimension m will be denoted by $\Omega^{[m]}$. It is easy to see that Ω^m is a subcomplex of Ω and $\Omega^{[m]}$ is not.

The *boundary* $\partial\gamma$ of a cell γ is defined to be the set of all proper subsets of γ. Cells ξ in $\partial\gamma$ are called *faces* of γ. Similarly, the *co-boundary* or *star* of a cell γ is defined as $\star\gamma = \{\xi \in \Omega \mid \gamma \subset \xi\}$. Cells ξ in $\star\gamma$ are called *co-faces* of γ. Any cell γ such that $\star\gamma = \{\gamma\}$ is called a *top cell* of Ω.

Two distinct cells are said to be *incident* if and only if one of them is a face of the other. Two simplices are called *s-adjacent* if and only if they share an

s-face. In particular, two p-simplices, with $p > 0$, are said to be *adjacent* if they are $(p-1)$-adjacent. Two vertices are called adjacent if and only if they are both incident at a common 1-cell. The *link* of a cell γ, denoted by $lk(\gamma)$, is the set of all faces of co-faces of γ, that are neither incident at, nor adjacent to γ.

A *h-path* is a sequence of simplices $(\gamma_i)_{i=0}^{k}$ such that two consecutive simplices in the sequence γ_{i-1} γ_i are h-adjacent. Two simplices γ and γ' are *h-connected* if and only if there exist a h-path such that γ is a face of γ_0 and γ' is a face of γ_k. A subset Ω' of a complex Ω is called *h-connected* iff every pair of its vertices are h-connected.

Classes of Complexes. A d-complex Ω in which every non-maximal simplex is a face of some maximal simplex is *regular* or *uniformly d-dimensional.*

A s-simplex γ in a d-complex, with $0 \leq s \leq d-1$, is a *manifold simplex* if and only if its link is combinatorially equivalent either to a $(d-s-1)$-sphere, or to a $(d-s-1)$-ball [7]. If γ is not a manifold simplex, it is called a *singularity.*

A regular $(d-1)$-connected d-complex where all $(d-1)$-simplices are manifold is called a *combinatorial pseudomanifold.* A regular d-complex where all vertices are manifold is called a *combinatorial d-manifold.* In a combinatorial manifold all simplices are manifold.

Topological Relations. Let γ be a p-simplex in a d-complex Ω, with $0 \leq p \leq d$. For each integer value q, $0 \leq q \leq d$, we define the topological relation $R_{pq}(\gamma)$ as a retrieval function that returns q-cells of Ω. Whenever $p < q$ function $R_{pq}(\gamma)$ returns the set of simplices of dimension q that contains γ. Similarly, for $p > q$, function $R_{pq}(\gamma)$ returns the set of simplices of dimension q that are contained in γ. Relation R_{pp}, for $p > 0$ is defined using R_{qp} for $q < p$, as $R_{pp}(\gamma) = \cup_{v \in \gamma} R_{(p-1)p}(\gamma - \{v\})$, i.e., $R_{pp}(\gamma)$ gives all p-simplices which are $(p-1)$-adjacent to γ. Similarly, $R_{00}(v) = \cup_{e \in R_{01}(v)}\{e - \{v\}\}$, i.e., $R_{00}(v)$ gives all 0-simplices which share a 1-simplex with v.

3 Related Work

Several data structures for manifolds can encode partially the non-manifold domain using simplicial and cell complexes.

Dimension-independent data structures have been proposed for d-dimensional manifold complexes, which include the *Cell Tuple (CT)* [1], the *n-G-map (nGM)* [12] for cellular complexes, and the *Indexed data structure with Adjacencies (IA)* for simplicial complexes (which directly extends to arbitrary dimension, being called *winged representation* in [15]).

If the IA is used to encode a simplicial d complex $2(d+1)$ references are needed for each d-simplex. If either CTs or n-G-maps are used to describe just simplicial complexes, they require $(d+1)!(d+1)$ references for each d-simplex. This represent a storage cost much bigger than that of IA, for a factor that grows combinatorially with the dimension of the complex.

The representation domain of all such data structures actually extends beyond the class of d-manifolds. The IA, altough extremely compact, can only describe Euclidean pseudomanifolds embedded in the Euclidean d-dimensional space. The n-G-maps describes a larger sub-class of pseudomanifolds introduced in [12], called *cellular quasi-manifolds*. The representation domain of CT is similar to that of n-G-maps (see [1] for details). However, none of them can encode completely the non-manifold domain.

A data structure for encoding any two-dimensional simplicial complex, called the *triangle-segment (TS)* data structure, has been proposed in [2]. The TS extends the IA to deal with non-manifoldness. This data structure is quite compact, since it requires at most $4n_{st}$ additional references with respect to the IA, where n_{st} denotes the number of top simplices incident to a non-manifold vertex. Moreover, the TS data structure downscale to IA in the manifold case.

Data structures for non-manifold, non-regular three-dimensional cell complexes have been proposed for modeling non-manifold solids. They are basically all variants of the *Radial-Edge (RE)* data structure [10]. The RE encodes any 3-cell implicitly through the manifold 2-complex partitioning its boundary. A face can be shared by at most two 3-cells. More compact versions of the RE, namely the *Tri-cyclic Cusps (TCC)* data structure ([8] and the *Partial Entity (PE)* data structure [11], have been proposed more recently. To give an idea of storage costs for these data strucutres we can compute the number of references necessary to encode a simplicial 3-complex by using such data structures. Let v, e, f, t be, respectively, the number of 0-, 1-, 2- and 3-cells/simplices in the non-manifold solid. Then, the RE uses $155t + 2f + e + v$ references, the TCC uses $94t + f + e + v$ references and finally storage requirements for PE reduces to $27t + 19f + 2e + v$ references. Experimental evaluations reported in [11] show that these data structures do not downscale well to the manifold case, i.e. they are extremely inefficient when used to encode manifolds.

In summary, we can conclude that data structures that fully models non-manifold solids do not downscales well to the manifold case. The data structure NMD-DS, presented in this paper, downscales effectively its storage requirements when going into the manifold domain. Still NMD-DS can encode a generic, possibly non-manifold, abstract simplicial d-complex. Of course, there are alternative ways for implementing abstract simplicial complexes, though not efficient. For instance *simplicial sets*, close to simplicial complexes, can be implemented as variants of incidence graphs [5], and chains of maps [6].

4 The Standard Decomposition

In this section, we summarize the results of previous work [3,14], in which we proposed a sound decomposition of non-manifold complexes.

We say that a decomposition Ω' is an *essential* decomposition of Ω if and only if all simplices of Ω' that must be pasted together to produce Ω are glued at some singularity (non-manifold face) of Ω.

The decompositions in Figures 1c and 1d are examples of essential decompositions for the complex of Figure 1a. The decompositions in Figures 1b and 1e

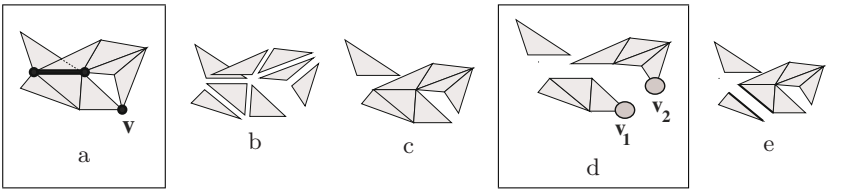


Fig. 1. A 2-complex with a non-manifold edge (having three incident triangles) and a non-manifold vertex V marked in bold (a); four possible decompositions of the complex (b).

are non-essential decompositions. The decomposition in Figure 1e is a manifold complex, but is not essential because we split along a manifold edge (marked in bold).

We consider essential decompositions as the only candidates, and we define the *standard* decomposition $\nabla\Omega$ as the most decomposed essential decomposition. It can be proven [14] that the standard decomposition exists, and it is unique and it is the decomposition that is obtained by cutting the complex Ω along *all* its non-manifold faces. For instance, the complex of Figure 1d is the standard decomposition of the complex in 1a. In [3], we have also presented an algorithm that computes the standard decomposition $\nabla\Omega$ in $O(d!\,t\log t)$ where t is the number of maximal simplices in the d-complex Ω. This decomposition algorithm produces a map, we will denote with σ, that maps back vertices in $\nabla\Omega$ into their original vertex in Ω (i.e. $\sigma(\nabla\Omega) = \Omega$). An example of the σ map associated with a standard decomposition is presented in Figure 3.

The standard decomposition is a complex formed of regular connected components, and each of its components belong to a class of complexes, that we called *initial quasi-manifolds*, which admit the local characterization.

A regular h-complex Ω is an *initial quasi-manifold* if and only if we can always traverse the maximal h-simplices in the star of each vertex through manifold $(h-1)$-faces (see [3] for the formal definition of initial quasi-manifolds). In this case we say that the star of each vertex is *manifold-connected*. This characterization is relevant to the design of data structures, as we will see in the next sections.

The class of initial quasi-manifolds coincides with that of manifolds in dimension $d \leq 2$, while in higher dimension ($d \geq 3$) there are initial quasi-manifolds which are non-manifold, and it also is possible to build examples of initial quasi-manifolds that are not even pseudomanifolds [14].

5 The Non-manifold Data Structure (NMD-DS)

In this section, we present a data structure, that we call NMD-DS, to encode non-manifold d-complexes according to their decomposition. The data structure contains a lower level, which encodes separately each initial quasi-manifold component obtained from decomposition; and an upper layer, which encodes

information necessary to traverse different components through (non-manifold) joints. The lower level is based on a data structure, we called the Initial Quasi Manifold Data Struture (IQM-DS) to encode initial quasi-manifolds that we sketched already in [4]. Next, in order to extract all topological relations efficiently , we add to this two level data structure a set of d partial relations, each denoted by V^iT, for $0 \leq i < d$. Partial relation V^iT will give, for each i-simplex γ, an h-simplex that is incident to γ. Proofs about claims and analyses of space and time complexity are omitted for brevity. All proofs can be found in [14].

5.1 The Initial Quasi Manifold Data Structure (IQM-DS)

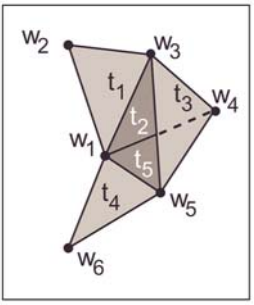
Let Ω be a h-dimensional initial quasi-manifold simplicial complex, with $h > 0$, we represent Ω with a data structure that extends the indexed data structure with adjacencies [15]. In the original IA, a complex is represented encoding relations R_{h0} and R_{hh} assuming that the encoded complex is a pseudomanifold. We extend this data structure to accomodate non-pseudomanifold situations where the set $R_{hh}(\xi)$ has two or more h-simplices. Let ξ be a non-manifold $(h-1)$-face, and let $\gamma_0, \ldots, \gamma_{k-1}$ be the h-simplices incident at ξ. Then for $i = 0 \ldots k-1$, in the adjacency list of γ_i, at the entry corresponding to ξ, we encode a link to $\gamma_{(i+1) \bmod k}$. This allows us to visit cells incident at ξ in cyclic order, thus supporting efficient retrieval of the complete relation R_{hh} for all such cells.

In Figure 2b we report references for the R_{20} and the R_{22} relations in the adjiacency data structure for the 2-complex in Figure 2a. As in the original IA symbol $\perp$ is used to mean "no adjacency". Note that in the tables for these two relations we adopt the usual consistency rules in ordering the two lists of links encoding R_{20} and R_{22} for a given triangle t, i.e.: the adjacency at a given position i in the list of R_{22} corresponds to the edge of t which is opposite to the vertex at the same position i in the list of R_{20}.

It is easy to see that, using this data strucure, we can encode R_{h0} and R_{hh} relations using $2(h+1)$ references for each h-simplex. Moreover through a suitable renumbering of vertices and top cells, we may obtain that a vertex indexed by w_i, for $i > h$, is always incident at a top h-cell indexed by $t_{(i-h)}$ (i.e. $w_i \in R_{h0}(t_{(i-h)})$ for $i > h$).

This renumbering allows us to maintain the V^0T implicitly. Furthermore, by imposing $R_{h0}(t_1) = \{w_1, \ldots, w_{h+1}\}$, and by exploiting $w_i \in R_{h0}(t_{(i-h)})$ for $i > h$ we can encode part of the R_{h0} relation implicitly saving v references. This reduce storage cost for the three relations R_{h0}, R_{hh} and V^0T relations $2f_h(h+1) - v$, where f_h is the number of h-simplices. The table in Figure 2, for instance, is built by using this numbering scheme (i.e. w_3 is in t_1, w_4 is in t_2 etc.).

Such a data structure is sufficient to retrieve all topological relations for an initial quasi-manifold h-complex. Vertex based relations R_{0m}, for $(h-2) \leq m \leq h$, can be computed in $O(|R_{0m}|)$ whenever the given abstract simplicial complex is imbeddable in $\mathbb{R}^h$. In particular we can compute in linear time all vertex bases topological relations R_{0m} in for 2 and 3 complexes embeddable in $\mathbb{R}^3$.

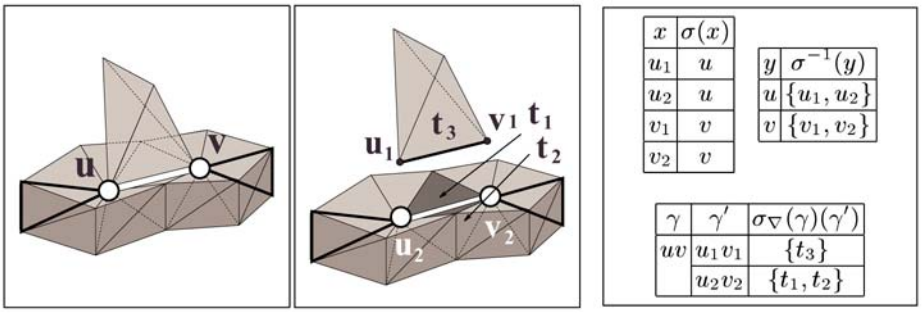


(a)

t	$R_{20}(t)$	$R_{22}(t)$
t_1	(w_1, w_2, w_3)	$(\perp, t_2, \perp)$
t_2	(w_4, w_1, w_3)	(t_1, t_3, t_5)
t_3	(w_5, w_4, w_3)	$(t_2, \perp, t_5)$
t_4	(w_6, w_5, w_1)	$(t_5, \perp, \perp)$
t_5	(w_5, w_1, w_4)	(t_2, t_3, t_4)

(b)

Fig. 2. References for the R_{20} and the R_{22} relations in the IA for the 2-complex on the left.



x	$\sigma(x)$
u_1	u
u_2	u
v_1	v
v_2	v

y	$\sigma^{-1}(y)$
u	$\{u_1, u_2\}$
v	$\{v_1, v_2\}$

γ	γ'	$\sigma_\nabla(\gamma)(\gamma')$
uv	u_1v_1	$\{t_3\}$
	u_2v_2	$\{t_1, t_2\}$

(a) (b) (c)

Fig. 3. A 3-complex (a) and its standard decomposition (b). Edge uv splits into u_1v_1 and u_2v_2 and the star of u_2v_2 is not manifold connected.

5.2 A Data Structure to Connect Components

We now assume that each component of $\nabla\Omega$ obtained from the decomposition algorithm is encoded with the IQM-DS data structure described in the previous subsection. In order to build a data structure for the original complex Ω, we add to this data structure the encoding of the map σ. Recall that the map σ is computed by the the decomposition process (See Section 4) and is such that $\sigma(\nabla\Omega) = \Omega$.

Consider for instance for the complex of Figure 3a whose standard decomposition is in Figure 3b. The maps σ and σ^{-1} for this complex are shown in Figure 3c.

The encoding of $\nabla\Omega$ together with maps σ and σ^{-1} is sufficient to extract all topological relations.

However, more complex relations are necessary to ensure more efficient traversal. To this aim, we introduce partial relation σ_∇. This relation is defined for all simplices $\gamma \in \Omega$ such that γ is either a *splitting* simplex or a simplex whose star is not $(h-1)$-connected. When γ is a splitting simplex, σ_∇ relates γ with the set

of simplices into which γ is split (they are called *copies* of γ). Figure 3c reports the map σ_∇ for the complex of Figure 3a. Note that edge uv splits into the two copies u_1v_1 and u_2v_2 and the star of u_2v_2 is not manifold-connected in $\nabla\Omega$.

The encoding of components of $\nabla\Omega$ together with the encoding of σ, σ^{-1} and σ_∇ is sufficient to compute efficiently all topological relations $R_{pq}(\gamma)$ provided that, we can supply a top simplex θ incident to γ. We will denote with $R_{pq}(\gamma|\theta)$ this computation (reads $R_{pq}(\gamma)$ given θ). If we assume logarithmic access time to the maps σ, σ^{-1} and σ_∇ then $R_{pq}(\gamma|\theta)$ can be computed, for a d-complex Ω imbeddable in $\mathbb{R}^d$, in $O(|R_{pq}(\gamma)| + \log n_{st})$ for all $(d-3) \leq p < q \leq d$, where n_{st} is the total number of top simplices incident to non-manifold vertices. This means that, for $d = 2$ and $d = 3$, under the above assumptions, all topological relations $R_{pq}(\gamma|\theta)$ are computed in $O(|R_{pq}(\gamma)| + \log n_{st})$.

Adding an encoding of relations V^pT we can provide a top simplex θ incident to a generic p-simplex γ. We assume that access to relation V^pT can be done in $O(\log |\Omega^{[p]}|)$ (recall that $\Omega^{[p]}$ is the set of all simplices of dimension p). With this assumption it is easy to see that relation $R_{pq}(\gamma)$ can be computed in $O(|R_{pq}(\gamma)| + \log n_{st} + \log |\Omega^{[p]}|)$.

5.3 Implementation and Storage Requirements

We describe here an implementation of the NMD-DS that is optimized for storage costs as well as for traversal operations. This implementation is inherently static and do not support editing operations.

Maps σ and σ^{-1} are encoded as balanced binary search trees, which support logarithmic access time, implemented as arrays.

Each entry in the array encoding map σ contains one key corresponding to a vertex copy and one pointer to its corresponding split vertex, for a total cost of $2n_c$ references, being n_c the number of vertex copies introduced by the decomposition process. Similarly each entry in the array encoding the map σ^{-1} contains one key corresponding to a split vertex and one pointer to the list of its vertex copies. All vertex copies can be maintained in a single array, segmented according to the different (disjoint) lists corresponding to split vertices. Thus, one list can be located in such array by two offset numbers, which can be compressed in a single reference. Therefore, σ^{-1} can be implemented by $2n_s + n_c$ references in total, being n_s the number of original vertices duplicated by the decomposition process ($n_c \geq 2n_s$).

Relations V^pT and σ_∇ are encoded as *trie dictionaries* whose words are the sequences of vertex indexes obtained by lexicographic ordering of simplices. A trie dictionary is usually implemented as a special binary search tree called a *ternary tree*. We assume again an array implementation of this tree. In this case, the trie for the map σ_∇ for a d-complex takes less than $(2^{d+1} - (d+3))n_{st}(4d+1)$ references.

In order to implement relations V^pT, for $0 \leq p < d$, we note that all trees for all tries for V^pT for all $0 \leq p < d$ overlap. From this property, assuming again an array implementation for the trie, the *collective* encoding of all relations V^pT for $0 < p < d$ can be done with $2|\Omega^{d-2}| + 2|\Omega^{[d-1]}| - v$ references.

Table 1. (a) Acronyms for data structures reviewed in Section 3; (b) ratios of storage costs against reviewed data structures for the NMD-DS used to encode a simplicial 3-manifold; (c) break-even thresholds on the number of singlarities that make other data structures more competitive than the NMD-DS.

(a)

IA	Indexed with Adjacencies [15]
CT	Cell Tuple [1]
nGM	n-G-Map [12]
RE	Radial Edge [10]
TCC	Tri-Cyclic Cusps [8]
PE	Partial Entity [11]
TS	Triangle-Segment [2]

(b)

	Ratio to NMD-DS
IA	< 0.21
CT	> 2.28
nGM	> 2.28

(c)

	threshold
RE	> 86
TCC	> 50
PE	> 33

6 Comparisons and Discussion

In this Section, we compare the NMD-DS data structure with the data structures reviewed in Section 3 and listed in Table 1a. In Table 1b we compare the NMD-DS, over the 3-manifold domain, against data structures IA, CT and nGM.

Over the 3-manifold domain the NMD-DS reduces to the IQM-DS augmented with the V^iT relations. Our comparison shows that the NMD-DS it requires nearly five times the space required by the IA. However, edge-based (R_{1h}) and face-based (R_{2h}) relations cannot be efficiently retrieved from the IA while it is possible to retrieve all topological relations in optimal time from the NMD-DS (see [14] for details).

The NMD-DS encodes non-manifoldness in a separate layer and thus NMD-DS storage requirements grow as the degree of the non-manifoldness increases. We have compared the NMD-DS with the RE, the TCC and the PE data structures used for representing non-manifold solids. For each of them, we compute a threshold, on the number of top simplices incident to a singular vertex, below which our data structure is more compact than the others. Table 1(c) summarizes the results of this analysis (see [14] for details). The break-even point, above which our data structure is no longer competitive, occurs, for the PE, when at least one third of top simplices are incident to a singular vertex. Storage cost of the NMD-DS and that of the TS both depends on the degree of non-manifoldness in the modeled 2-complex. However, under the hypthesis that the average vertex order is greater than six we find that the TS is always more compact than the NMD-DS.

7 Concluding Remarks

In this paper, we have introduced a new, dimension-independent, data structure for describing simplicial complexes, called the Non-Manifold Decomposition data structure (NMD-DS). The NMD-DS is a two-level data structure being based

on a decomposition of the complex into simpler components, called initial quasi-manifolds, which can be encoded in a compact data structure supporting efficient traversal. We have reviewed and analyzed existing data structures for simplicial and cell complexes, and we have evaluated the NMD-DS data structure with respect to them.

The NMD-DS structure supports efficient traversal algorithms, and it is compact. In particular, our analysis has shown that it is more compact than any data structure for non-manifold solids when less than one third of the cells of the complex are non-manifold. Moreover, the NMD-DS structure scales very well to the manifold case, since it exhibits a negligible overhead when applied to a manifold complex.

Acknowledgements

This work has been performed while Leila De Floriani has been visiting the Computer Science Department of the University of Maryland at College Park (USA). This work has been supported by the Italian Ministry of Education University and Research under FIRB project MACROGeo (contract N.RBAU01MZJ5), by the Italian Space Agency (ASI) under project "Augmented Reality for Teleoperation of Free Flying Robots" and by the Italian National Research Council under research project *Efficient modeling and transmission of three-dimensional scenes and objects* under program "Agenzia 2000", Contract N.CNRC00FE45_004.

References

1. E. Brisson. Representing geometric structures in d dimensions: Topology and order. In *Proceedings 5th ACM Symposium on Computational Geometry*, pages 218–227. ACM Press, June 1989.
2. L. De Floriani, P. Magillo, E. Puppo, and D. Sobrero. A multi-resolution topological representation for non-manifold meshes. In *Proceedings 7th ACM Symposium on Solid Modeling and Applications (SM02) Saarbrucken, Germany, June 17-21*, pages 159–170, 2002.
3. L. De Floriani, M.M. Mesmoudi, E. Puppo, and F. Morando. Non-manifold decomposition in arbitrary dimensions. In A. Vialard A. Braquelaire, J.O. Lachaud, editor, *Discrete Geometry for Computer Imagery*, volume 2301 of *Lecture Notes in Computer Science*, pages 69–80. Springer-Verlag, 2002. Extended version to appear in *Graphical Models*.
4. L. De Floriani, F. Morando, and E. Puppo. Representation of non-manifold objects through decomposition into nearly manifold parts. In V. Shapiro G. Elber, editor, *Proceedings 8th ACM Symposium on Solid Modeling and Applications Seattle, WA, June 16-20*, pages 304–309. ACM Press, 2003.
5. H. Edelsbrunner. Algorithms in combinatorial geometry. In Brauer, W., Rozenberg, G., and Salomaa, A., editors, *EATCS Monographs on Theoretical Computer Science*. Springer-Verlag, 1987.
6. H. Elter and P. Lienhardt. Cellular complexes as structured semi-simplicial sets. *International Journal of Shape Modeling*, 1(2):191–217, 1994.

7. L. C. Glaser. *Geometric combinatorial Topology.* Van Nostrand Reinhold, New York, 1970.
8. E. L. Gursoz, Y. Choi, and F. B. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. U. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 107–130. Elsevier Science Publishers B.V., North Holland, 1990.
9. V. A. Kovalevsky. Finite topology as applied to image analysis. *Computer Vision, Graphics, and Image Processing*, 46(2):141–161, May 1989.
10. K.Weiler. The radial edge data structure: A topological representation for non-manifold geometric boundary modeling. In H.W. McLaughlin J.L. Encarnacao, M.J. Wozny, editor, *Geometric Modeling for CAD Applications*, pages 3–36. Elsevier Science Publishers B.V. (North–Holland), Amsterdam, 1988.
11. S.H. Lee and K. Lee. Partial entity structure: a fast and compact non-manifold boundary representation based on partial topological entities. In *Proceedings Sixth ACM Symposium on Solid Modeling and Applications.* Ann Arbor, Michigan, June 2001.
12. P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *CAD*, 23(1):59–82, 1991.
13. M. Mantyla. *An Introduction to Solid Modeling.* Computer Science Press, 1983.
14. F. Morando. *Decomposition and Modeling in the Non-Manifold domain.* PhD thesis, February 2003.
15. A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci. Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics*, 12(1):56–102, January 1993.