Research Summary

David Canino

E-mail: canino.david@gmail.com
Website: http://mangrovetds.sourceforge.net/pages/david_en.php

Summary. In this document, I summarize briefly results I have obtained in my academic career, in particular during my PhD. and Postdoc Studies in Computer Science at the Department of Computer Science, University of Genova (GE), Italy, under the supervision of Professor Leila De Floriani.

1 Introduction

In my PhD. research [6], I have addressed mainly the efficient representation of *non-manifold shapes*. Informally, a *manifold shape* is a subset of the Euclidean space, such that every neighborhood of each point is locally equivalent to an open ball. Shapes, that do not fulfill this property at one or more points, are called *non-manifolds*. Points, that do not satisfy the manifold property, are known as *non-manifold singularities*. The use of non-manifold shapes is common in many applications, like the numerical simulations [28]. In particular, they provide more accurate representations of real objects, usually formed by several components of different dimension, arbitrarily connected.

Modeling of non-manifold shapes requires *efficient representations* for their discretized versions through simplicial and cell complexes, and *topological methods* for analyzing their structure. In this context, the objective of my research has been twofold. On one side, I have investigated effective representations for modeling non-manifold shapes, discretized as abstract simplicial and cell complexes of any dimension, not necessarily embedded in any Euclidean space. On the other hand, I have considered a *topological decomposition* of any non-manifold shape into manifold or almost manifold parts, connected through non-manifold singularities, in order to deal with their intrinsic complexity. This topological decomposition is also the basis for computing iteratively the simplicial homology of any non-manifold shape.

2 Representing Shapes by Topological Data Structures

Following [16], *topological relations* describe connectivity information for each topological entity. They are the basis for *topological data structures*, described in terms of what entities and what relations they encode. In this context, I have provided two new topological data structures, specific for non-manifold shapes of arbitrary dimension, discretized by abstract complexes.

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 do not depend on the embedding domain. The *Incidence Graph (IG)* [19] is one of the most common dimension-independent data structures for representing non-manifold shapes, discretized by simplicial and cell complexes, and not necessarily embedded in any metric space. The IG data structure encodes all cells of the complex, as well as a subset of incidence relations among cells. The IG data structure is verbose, and exhibits a very large overhead, if restricted to manifolds. It does not allow detecting non-manifold singularities efficiently, since it provides no information regarding the structure for each neighborhood of any cell γ. This latter must be reconstructed when it is necessary, and the complexity of this operation is linear in the number of cells, incident at γ. Thus, more compact representations are needed, from which it is efficient to detect non-manifold singularities.

In this context, I have proposed the *Incidence Simplicial (IS)* data structure [17], and *Generalized Indexed data structure with Adjacencies (IA*)* [13]. I have also proposed the *Mangrove Topological Data Structure* framework [12], focused on the fast design of topological data structures under the same Application Programming Interface (API) in the same spirit of the rapid prototyping techniques in the manufacturing [30].

The *Incidence Simplicial (IS)* data structure [17] is a variant of the IG data structure for representing abstract complexes of any dimension. It encodes all cells in the complex, and a subset of incidence relations among these cells. In particular, for any cell γ, it encodes immediate faces of γ (like in the incidence graph), and a subset of immediate cofaces, one for each connected component in the neighborhood of γ. The IS data structure is more compact than the incidence graph, and it exhibits a small overhead, if restricted to manifolds [6]. It provides information regarding the structure of the neighborhood of any cell, and this information is exploited for recognizing non-manifold singularities efficiently. The IS data structure is suitable for applications, i.e., numerical simulations [28], where one needs to attach attributes to all cells.

Incidence-based data structures offer a complete description of complexes, but more compact representations can be obtained, for instance when the specific application does not require attaching attributes to all cells. Following [16], *adjacency-based* data structures encode only vertices and top cells, i.e., cells not on the boundary of other cells, as well as adjacency relations, restricted to top cells, and co-boundary relations for vertices. Existing representations are only for manifold complexes [14], or for non-manifold shapes, necessarily embedded in the Euclidean 3D space [15, 18].

In order to overcome these limitations, I have designed the *Generalized Indexed Data Structure with Adjacencies (IA*)* [13]. The IA* data structure is an adjacency-based data structure for abstract complexes of any dimension, not necessarily embedded in the Euclidean space. In particular, it encodes all the vertices and top cells, plus a subset of co-boundary relations for vertices, and adjacency relations, restricted to top cells. It provides information regarding the structure of the neighborhood for any vertex, and of any $(p-1)$-cell, shared by more than two top p-cells. This information can be exploited for recognizing non-manifold singularities efficiently, namely in constant time. The IA* data structure is more compact than incidence-based representations [6], and of dimension-specific ones [15,18]. It also supports efficiently the retrieval of all the topological relations, involving cells, that are directly encoded. The efficiency of other topological queries, involving cells that are not directly encoded, depends on the implicit representations of cells of interest.

As discussed in [16], several topological data structures have been introduced in the literature. Hence, a framework, which supports a wide number of topological data structures under a common application interface would be a valuable tool from the applicative point of view. In particular, such a framework should be focused on the fast design of topological data structures in the same spirit of the *fast prototyping* techniques in the manufacturing [30]. The key idea consists of creating a *prototype* of any topological data structure, which can be customized in order to simulate a specific data structure without introducing any overhead. In this way, the internal representation of the complex can be *extended* easily. Each topological data structure is seen as a *plugin* to be loaded dynamically in the system in order to choose the most efficient representation for any specific task. In any case, a framework with these properties is currently lacking in the literature [27].

In order to overcome this lacking, I have designed the *Mangrove Topological Data Structure (Mangrove TDS)* framework [12]. Here, the connectivity information among cells in any complex is described as a graph, that is a *mangrove* in the most cases. Mangroves, that form a particular class of graphs, are thus exploited for describing the generic prototype of a topological data structure. A data structure is described by its corresponding mangrove. This allows to reuse several theoretical results of graphs theory, applied to topological representations, e.g., topological queries are expressed as breadth-first traversals of mangroves. This implies that the internal representation of this framework, able to represent graphs, can be dynamically replaced by exploiting the most suitable mangrove for the task of interest. Note that this can be done under a unique API, which hides the content of any mangrove, giving the possibility to write a program only once, and to perform coherent comparisons regarding the performance of topological queries among different representations.

Another interesting feature of the framework is provided by what I call the *ghost topological entities*. Following [16], adjacency-based data structures encode only vertices and top cells, i.e., cells that do not bound other cells. As a consequence, it is not possible (or it is not efficient) to execute topological queries on any cell, if not directly encoded. In particular, this depends not only on the properties of the data structure, but also on the representations of cells, that are not encoded. Thus, implicit representations of these cells are necessary, and most of implicit representations in the literature are mostly for manifold complexes. Specifically, ghost topological entities are implicit representations of those cells, not encoded explicitly, in any non-manifold and not pure domain (formed by pieces of different dimension), which is approximated by a cell complex of any dimension. The key idea consists of associating any cell γ, not necessarily encoded (called the *child cell*), with at least one arbitrary top cell γ' (called the *parent cell*), bounded by the child cell γ. Thus, it is possible to exploit adjacency relations, encoded only for top cells, without navigating on a maximal path from child cell γ to top cell γ' in the incidence-graph (i.e., Hasse diagram), and in similar representations. Experiments in [6] show that these representations improve the efficiency of topological queries, and that any adjacency-based representation becomes equivalent to any incidence-based representation, in which all cells are encoded. Ghost topological entities require auxiliary data structures for being effective. In any case, this overhead does not depend on the size of the complex, but only on the types of cells, and it is negligible. Thus, adjacency-based data structures, if equipped with ghost topological entities, are a fair solution for many tasks with respect to incidence-based data structures, that are considerably more expensive.

In order to prove the validity of this approach, I have designed and implemented several data structures with complementary properties, including the IS and IA* representations. I have provided navigation and construction algorithms for these topological data structures. I have also performed quantitative comparisons regarding their performances [6]. The complete implementation of the Mangrove TDS framework, including all the data structures, is contained in the *Mangrove Topological Data Structure (Mangrove TDS) Library*, which is released in public domain for the community [8, 9, 11].

3 Topological Decomposition of Shapes

Another way to represent a non-manifold shape consists of decomposing the shape of interest into manifold (or almost manifold) parts, in order to deal with its intrinsic complexity. In this context, a decomposition of any non-manifold shape should remove as many singularities as possible, and cut the shape along singularities without breaking it at manifold parts. In my research, I have considered the *Manifold-Connected (MC-) decomposition.* The basic concepts underlying this decomposition, but limited to 2D and 3D simplicial complexes, have been introduced in [20, 21]. Here, any non-manifold shape of dimension d is decomposed into nearly-manifold components of dimension d, known as the *Manifold-Connected (MC-) components.* In particular, any MC-component of dimension d is defined by the maximal set of top d-cells, that can be reached iteratively from any top d-cell γ by navigating over adjacency relation along any $(d-1)$-faces τ, such that at most two top d-cells are incident at τ. The MC-decomposition is unique, and it is the discrete counterpart of the Whitney stratification [29]. In particular, the MC-decomposition of any non-manifold shape Γ exposes the *structure* of Γ, its meaningful components, and their connectivity along non-manifold singularities. Thus, the MC-decomposition provides a *structural model* of any non-manifold shape Γ.

A suitable representation for the MC-decomposition is given by a graph-based data structure, where a node corresponds to one MC-component, while an arc (possibly an hyperarc) describes the connectivity among several MC-components, connected through a subcomplex, formed by non-manifold singularities. The difference among graph-based representations is given by the different intersection of MC-components to be considered. Specifically, I have designed and implemented three two-level graph-based representations of the MC-decomposition, which are combined with any topological data structure, as discussed in [7, 10]. The upper level of these graph-based data structures consists of a collection of MC-components, while the lower level consists of an unique topological data structure. In this context, each MC-component is completely defined in terms of references to cells in the input shape, in the same spirit of spatial indices [23]. I have implemented these graph-based data structures by combining them with all representations in the Mangrove TDS Library. This solution show the validity of my approach, since, combinatorial aspects, related to mangroves, are completely separated from structural aspects, related to the MC-decomposition. In particular, I obtain structural models, based not only on representations encoding all cells, e.g., the IS data structure, but also on representations, e.g., the IA* data structure, that encode only a subset of cells. I have evaluated the storage costs and the building times for all topological data structures in the library. Tests in [10] show that these graph-based data structures are more expressive than a topological data structure, since they explicitly expose singularities and connectivity of MC-components. In particular, if combined with the IA* data structure, they offer a structural model, which does not introduces overhead with respect the data structure of interest, when representing a manifold. The resulting representation is even more compact than the incidence graph, which is verbose and does not expose non-manifold singularities explicitly.

Recently, computing any topological invariants of a shape has drawn much attention, because they provide any information, which is very useful when pure geometric tools are not sufficient. *Simplicial homology* provides the most common topological invariants, as shown in [2]. In my research, I have addressed the problem of computing simplicial homology of any non-manifold shape by an iterative algorithm. Classical techniques for computing the simplicial homology exploit an algebraic approach [1,22], which does not yield to an iterative algorithm [24]. Instead, the *Constructive Homology Theory* [24–26] offers an elegant way for iteratively computing the homology of a shape from the homology of its subcomplexes and of their intersections. This approach is based on a generic decomposition of any shape, even if it is manifold. In my research, I have exploited this approach in order to design the *Mayer-Vietoris (MV)* algorithm [3], which retrieves homological information from a decomposition of any non-manifold shape. I have combined the MV algorithm with the MC-decomposition of non-manifold shapes. This algorithm is dimension-independent and always decidable. In this case, the MV algorithm relates the simplicial homology of MC-components, and of the intersection of any pair of MC-components. In this case, the intersection of two MC-components is given by a subcomplex of non-manifold singularities, whose size is usually limited. I have experimentally shown that the MC-decomposition increases the efficiency of the MV algorithm with respect to the classical approaches.

4 Out-of-Core Representations

Another secondary field of interest in my previous research is given by the analysis of *out-of-core representations*. Nowadays, performances of graphics subsystems has enormously improved, but unfortunately the complexity of graphics applications has also increased. Some huge representations can be easily produced in several applications, for example by the 3D scanning of real objects, or in medical applications. In order to improve the *resolution* of a representation, accurate object sampling is needed, and, thus, the size of a representation increases, and it often exceeds the amount of RAM in a workstation. Hence, these representations introduce severe overheads, and their management has often prohibitive costs, even for high-performance workstations. Thus, an *out-of-core* technique is mandatory. In this context, the entire shape is maintained in the *External Memory (EM)*, and only portions of interest, small enough to be processed in-core, are dynamically loaded in RAM. In this way, it is possible to operate directly on each portion by removing limitations on the input model size. In any case, an EM access is slower than a RAM access: if an efficient control of EM accesses is not performed, then this fact is a bottleneck, and it degrades performance. An excellent solution to this problem is given by the decomposition of shapes through *spatial indices* [23].

In the same spirit of the Mangrove TDS framework, I have proposed the *Objects Management in Secondary Memory (OMSM)* framework [4, 5]. The OMSM framework is a general-purpose framework for managing an unstructured and huge set of spatial objects. Here, a generic storage architecture is described in terms of three aspects, namely the *space partitioning tree*, the *clustering policy* of nodes in a spatial index, grouped in atomic units, known as the *clusters*. Finally, a cluster is transferred between any *storage support* and RAM. Each of these aspects does not depend on each other, and is described by a *dynamic plugin*, including the represented objects. Hence, the OMSM framework may be easily adapted to the users needs through dynamic plugins, providing many techniques to be integrated in a storing architecture. A new technique is made available in the OMSM framework without modify the framework.

References

1. Max K. Agoston. *Algebraic Topology: a First Course.* M. Dekker Publisher, 1976.
2. Max K. Agoston. *Computer Graphics and Geometric Modeling: Mathematics.* Springer, 2005.
3. Dobrina Boltcheva, David Canino, Sara Merino Aceituno, Jean-Claude Léon, Leila De Floriani, and Franck Hétroy. An Iterative Algorithm for Homology Computation on Simplicial Shapes. *Computer-Aided Design*, 43(11):1457–1467, 2011.
4. David Canino. An Extensible Framework for Huge Geometric Models. Technical Report DISI–TR–09–08, Department of Computer Science, Genova, Italy, December 2009.
5. David Canino. A Dimension-Independent and Extensible Framework for Huge Geometric Models. In Enrico Puppo, Andrea Brogni, and Leila De Floriani, editors, *Proceedings of the 8th Eurographics Italian Chapter Conference*, pages 111–116. Eurographics Association, 2010.
6. David Canino. *Tools for Modeling and Analysis of Non-manifold Shapes.* PhD thesis, Department of Computer Science, Bioengineering, Robotics and Systems Engineering (DIBRIS), University of Genova, Genova, Italy, 2012.
7. David Canino and Leila De Floriani. A Decomposition-based Approach to Modeling and Understanding Arbitrary shapes. In Andrea Abate, Michele Nappi, and Genny Tortora, editors, *Proceedings of the 9th Eurographics Italian Chapter Conference*, pages 53–60. Eurographics Association, 2011.
8. David Canino and Leila De Floriani. *The Mangrove Topological Data Structure Library*, 2012. `http://mangrovetds.sourceforge.net`.
9. David Canino and Leila De Floriani. An Extensible Framework for Modeling Simplicial Complexes. In *Poster Compendium of the IEEE Visualization*, 2012.
10. David Canino and Leila De Floriani. A Compact Representation for Topological Decompositions of Non-manifold Shapes. In *8th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 100–107, 2013.
11. David Canino and Leila De Floriani. An Extensible Framework for Modeling Simplicial and Cell Complexes. In *Poster Compendium of the 11st Eurographics Symposium on Geometry Processing*, 2013.
12. David Canino and Leila De Floriani. Representing Simplicial Complexes with Mangroves. In *Proceedings of the 22nd International Meshing Roundtable*, pages 465–483. Springer, 2013.
13. David Canino, Leila De Floriani, and Kenneth Weiss. IA*: an Adjacency-based Representation for Non-Manifold Simplicial Shapes in Arbitrary Dimensions. *Computer & Graphics*, 35(3):747–753, 2011.
14. Leila De Floriani. *Basic Notions of Combinatorial Topology*, 2003. Lectures Notes, Master's Course on Geometric Modeling.
15. Leila De Floriani and Annie Hui. A Scalable Data Structure for Three-Dimensional Non-manifold Objects. In *Proceedings of the 1st Eurographics Symposium on Geometry Processing*, pages 77–82. ACM Press, 2003.
16. Leila De Floriani and Annie Hui. Data Structures for Simplicial Complexes: an Analysis and a Comparison. In *Proceedings of the 3rd Eurographics Symposium on Geometry Processing*, pages 119–128. ACM Press, 2005.

17. Leila De Floriani, Annie Hui, Daniele Panozzo, and David Canino. A Dimension-Independent Data Structure for Simplicial Complexes. In *Proceedings of the 19th International Meshing Roundtable*, pages 403–420. Springer, 2010.
18. Leila De Floriani, Paola Magillo, Enrico Puppo, and Davide Sobrero. A Multi-resolution Topological Representation for Non-Manifold Meshes. *CAD Journal*, 36(2):141–159, 2004.
19. Herbert Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer, 1987.
20. Annie Hui and Leila De Floriani. Topological Decompositions for 3D Non-manifold Simplicial Shapes. Technical Report CS-TR-4855/UMIACS-TR-2007-10, Department of Computer Science, University of Maryland at College Park, MD, USA, 2007.
21. Annie Hui and Leila De Floriani. A Two-level Topological Decomposition for Non-Manifold Simplicial Shapes. In *Proceedings of the ACM Symposium on Solid and Physical Modeling*, pages 355–360. ACM Press, 2007.
22. James Munkres. *Algebraic Topology*. Prentice Hall, 1999.
23. Hanan Samet. *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann, 2006.
24. Francis Sergeraert. The Computability Problem in Algebraic Topology. *Advances in Mathematics*, 104(1):1–29, 1994.
25. Francis Sergeraert. Constructive Algebraic Topology. *ACM SIGSAM Bulletin*, 33(3):13–25, 1999.
26. Francis Sergeraert and Julio Rubio. *Constructive Homological Algebra and Applications*, 2006. urlhttp://www-fourier.ujf-grenoble.fr/ sergerar/Papers/.
27. Daniel Sieger and Mario Botsch. Design, Implementation, and Evaluation of the Surface Mesh Data Structure. In *Proceedings of the 20th International Meshing Roundtable*, pages 533–550. Springer, 2011.
28. Atul Thakur, Ashis Gopal Banerjee, and Satyandra Gupta. A Survey of CAD Models Simplification Techniques for Physics-based Simulation Applications. *Computer-Aided Design*, 41(2):65–80, 2009.
29. Hassler Whitney. Local Properties of Analytic Varieties. In Stewart Cairns, editor, *Differential and Combinatorial Topology - A Symposium in Honor of Marston Morse*, pages 205–244. Princeton University Press, 1965.
30. Paul Kenneth Wright. *21st Century Manifacturing*. Prentice Hall, 2001.