A Decomposition-based Approach to Modeling and Understanding Arbitrary Shapes

David Canino [1] and Leila De Floriani [1]

[1] Department of Computer Science, Universitá degli Studi di Genova, Italy

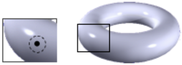
9th *Eurographics Italian Chapter Conference*, November 24-25, 2011
Salerno (SA), Italy

November 24, 2011

Arbitrary Shapes

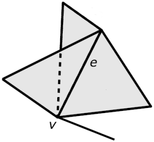
Manifold shapes

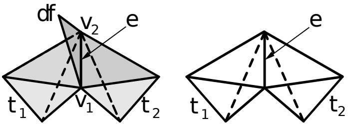
Each point has a neighborhood homeomorphic to either an open ball (*internal point*), or to a closed half-ball (*boundary point*).

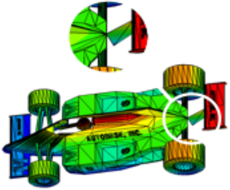


Arbitrary shapes (non-manifold / non-regular)

- *non-manifold singularities*, i.e. points at which the manifold condition is not satisfied;
- parts of *different dimensions*.





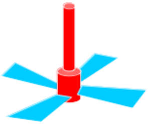


What we propose

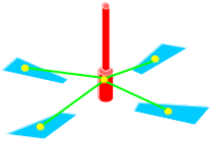
Motivation

Complex topology of an arbitrary shape offers *valuable information* to:

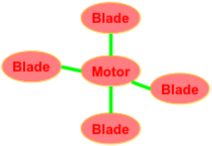
- shape annotation and retrieval, identification of form features;
- computation of $\mathbb{Z}$-homology (generators, Betti numbers, torsion coefficients).



Topological data structure



Structural model (shape decomposition)



Semantic model (future work)

Our proposal (Manifold-Connected Decomposition)

A structural representation based on topological aspects (*manifold-connected*).

Data Structures for Arbitrary Shapes

Context

- A lot of data structures for manifold simplicial and cell complexes
- Very few for arbitrary simplicial and cell complexes

Related Work

- *Incidence Graph* [Edelsbrunner 1987]: a dimension-indepedent data structure for arbitrary cell complexes, and restrictions to simplicial complexes, *Incidence Simplicial [De Floriani, Hui, Panozzo, Canino, 2010]*
- Representations for arbitrary 2D shapes in 3D: from *Radial Edge [Weiler, 1985]* to *Partial Entity [Lee and Lee, 2003]*;
- Dimension-specific data structures for 2D and 3D simplicial shapes;
- Representations for cell 2-complexes (decomposition into manifold parts).

Our Contribution [Canino, De Floriani, Weiss 2011, *SMI Conf. 2011*]

Generalized Indexed Data Structure with Adjacencies (IA^* data structure)

Representation of IA* [Canino et. al. 2011]

Entities

- Vertices;
- *Top simplices* (not on the boundary of any simplex).

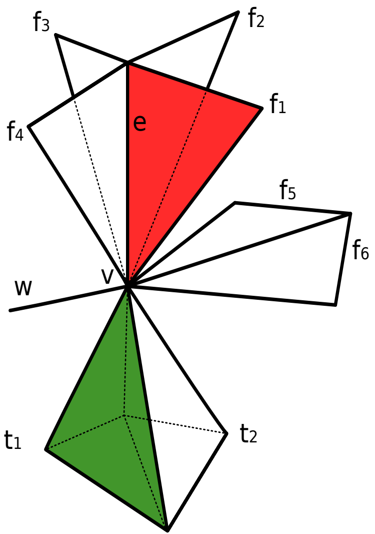
Encoded Relations

- $\mathcal{R}^*_{k,0}$ - vertices of top k-simplices;
- $\mathcal{R}^*_{0,k}$ - one top k-simplex for each $(k-1)$-connected component of simplices incident at a vertex;
- $\mathcal{R}^*_{k,k}$ - adjacency relation for top k-simplices, $k > 1$;
- $\mathcal{R}^*_{k-1,k}$ - partial co-boundary relation for non-manifold $(k-1)$-simplices incident to top k-simplices.

Properties

- Adiacency-based Representation;
- Dimension-independent;
- Arbitrary shapes;
- *Agnostic* about embedding in underlying space;
- *Scalable* with respect to manifold case (reduces to IA);
- Efficient retrieval of *topological relations*;
- Supports *editing* operations;
- *Most compact* encoding, with respect to the start of the art.
- Plan to release as part of C++ open source meshing library *Mangrove TDS*.

IA* data structure - an Example



$$\mathcal{R}^*_{0,1}(v) = \{w\}$$
$$\mathcal{R}^*_{0,2}(v) = \{f_1, f_5\}$$
$$\mathcal{R}^*_{0,3}(v) = \{t_1\}$$
$$\mathcal{R}^*_{2,2}(f_5) = \{f_6\}$$
$$\mathcal{R}^*_{2,2}(f_6) = \{f_5\}$$
$$\mathcal{R}_{3,3}(t_1) = \{t_2\}$$
$$\mathcal{R}^*_{2,2}(f_1) = \mathcal{R}^*_{2,2}(f_2) = \mathcal{R}^*_{1,2}(e)$$
$$\mathcal{R}^*_{2,2}(f_3) = \mathcal{R}^*_{2,2}(f_4) = \mathcal{R}^*_{1,2}(e)$$

$$\mathcal{R}^*_{1,2}(e) = \{f_1, f_2, f_3, f_4\}$$

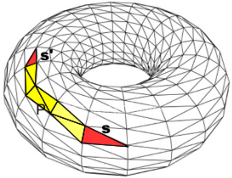
Key observation

- Encode collection of top k-simplices incident to a non-manifold $(k-1)$-simplex as a *single unit* and once.
- Efficient retrieval of non-manifold *singularities*.

Manifold-Connected (MC) Components

Manifold k-Path

Sequence of k-simplices, where each pair of simplices is *adjacent* through a *manifold* $(k-1)$-simplex.

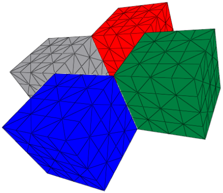


Manifold-Connected (MC) k-simplices

Connected through a manifold k-path.

Manifold-Connected (MC) Complex

All pairs of MC k-simplices.



Key property

Unique if and only if we consider *top simplices*.

Retrieving MC-components

Algorithm 1 RETRIEVE_MC_COMPONENTS(Σ)

Input: an instance of the IA* data structure representing Σ
Output: the set of non-manifold singularities in Σ and their related MC-components

let $L := \emptyset$
for all $k = 1, \ldots, dim(\Sigma)$ **do**
  for all top k-simplex σ in Σ **do**
    if σ is not visited **then**
      create a new MC-component $\mathcal{C}$
      let q an empty queue
      enqueue σ in q
      while q is not empty **do**
        dequeue σ' from q
        if σ' is not visited **then**
          mark σ' as visited
          add σ' in the new MC-component $\mathcal{C}$
          for all τ in $b(\sigma')$ **do**
            if τ is not manifold in Σ **then**
              $L[\tau] := L[\tau] \cup \{\mathcal{C}\}$
            else if $dim(\tau) = k - 1$ **then**
              enqueue $\mathcal{R}^*_{k,k}(\sigma')$ along τ in q
            end if
          end for
        end if
      end while
    end if
  end for
end for
return L

Basic idea

Recognize MC-components for each sub-complex in Σ formed by top k-simplices.

Basic step

Proceed by adiacency on manifold $(k-1)$-faces of a top k-simplex σ.

At the end

- Each top simplex associated to ONE MC-component;
- Singularities associated to several MC-components.

Time complexity

Linear in the number of simplices in Σ.

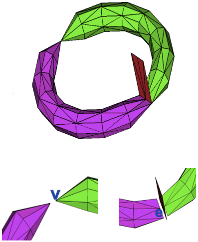
Manifold-Connected (MC) Decomposition

MC-Decomposition

- Collection of *MC-components* in the input arbitrary shape Ψ;
- Discrete counterpart of the *Whitney stratification* (1965);

MC-components

- *equivalence classes* of top simplices in Ψ vs MC relation;
- share *non-manifold singularities*;
- a singularity may be shared by *more than one* MC-component.



Consequence

Suitable to be represented through a *two-level graph-based data structure.*

Representing the MC-Decomposition

Two-level Data Structure

- the *lower level* describes an arbitrary shape Ψ through an unique IA* data structure (*topological model*);
- the *upper level* describes the connectivity of MC-components in Ψ through a graph-based data structure (*structural model*).

MC-graph $(\mathcal{N}, \mathcal{A})$

- each *node* in $\mathcal{N} \equiv$ one MC-component (direct references to simplices in Ψ);
- each *arc* in $\mathcal{A} \equiv$ *intersection* (non-manifold singularities) between two or more MC-components;
- similar to a *spatial index* overimposed on Ψ.

Variants of the MC-graph (encodings of arcs)

- *Pair-wise MC-graph* (intersection of two MC-components);
- *Extended MC-graph* (intersection of more than two MC-components).

Pair-wise MC-graph $\mathcal{G}_P = (\mathcal{N}_P, \mathcal{A}_P)$

Key Property

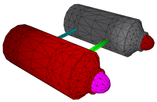
Arc $\equiv$ intersection of only two MC-components

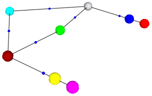
Encoding a node c

- the dimension k of top simplices in c;
- s_c^p references to top k-simplices in c;
- a_c^p references to arcs incident in c.

Encoding an arc $a = (c_1, c_2)$

- two references to c_1 and c_2;
- l_a^p references to singularities in $c_1 \cap c_2$.





CCViewer, L. De Floriani, D. Panozzo, A. Hui, GbPR 2009

Storage cost

$$\sum_{c \in \mathcal{N}_P} (1 + s_c^p + a_c^p) + \sum_{a \in \mathcal{A}_P} (4 + l_a^p)$$

Extended MC-graph $\mathcal{G}_E = (\mathcal{N}_E, \mathcal{A}_E)$

Key Property

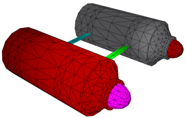
One *hyper-arc* for each singularity σ common to any MC-components

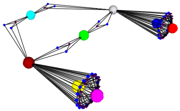
Encoding a node c

- the dimension k of top simplices in c;
- s_c^e references to top k-simplices in c;
- a_c^e references to arcs incident in c.

Encoding an arc connecting any MC-components

- a reference to the singularity σ related to a;
- l_σ references to MC-components sharing σ.





CCViewer, L. De Floriani, D. Panozzo, A. Hui, GbPR 2009

Storage cost

$$\sum_{c \in \mathcal{N}_E} (1 + s_c^e + a_c^e) + \sum_{a \in \mathcal{A}_E} (1 + l_\sigma)$$

Comparisons I

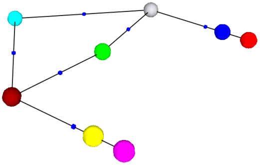
Key observation

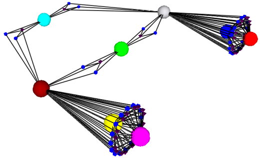
These graph-based data structures *adapt themselves* to the shape complexity:

- simplices in the *intersection* between two MC-components;
- *number of MC-components* incident at a singularity.

In the most cases...

The Extended MC-graph is 30% smaller than the Pair-wise MC-graph, but...

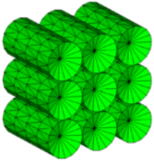




Comparisons II

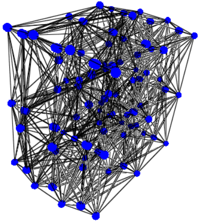
Storage Cost (vs IG)

- Extended MC-graph + IA* $\equiv$ 78% of IG in 2D;
- Pairwise MC-graph + IA* $\equiv$ 86% of IG in 2D;
- both of them + IA* $\equiv$ 35% of IG in 3D



Storage Cost (vs IA*)

- both of them are about 40% of the IA* in 2D;
- both of them are about 38% of the IA* in 3D.
- both of them expose singularities and connectivity of MC-components.



In Boltcheva et al. 2011 (GD/SPM 2011)

- the size of intersection between MC-components does not exceed 5% of the input shape;
- the dimension of a MC-component is on average 40% of the input shape.

Iterative Computing of $\mathbb{Z}$-homology [Boltcheva et al. 2011]

Objective

Computation of $\mathbb{Z}$-homology for an arbitrary shape:

- Constructive Homology Theory [Sergeraert, 2006];
- MC-decomposition.

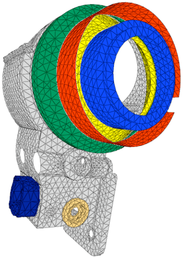
Basic idea

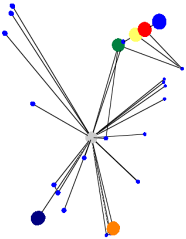
Compute the homology of a shape by combining:

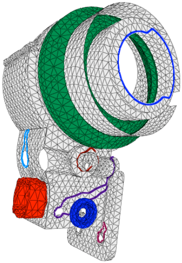
- homology of its sub-complexes;
- homology of the intersection of sub-complexes.

Results

Better than the SNF algorithm.



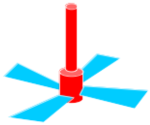




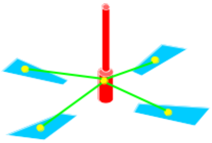
Conclusions and Future Work

We have presented the *MC-Decomposition*, an effective structural model for several applications:

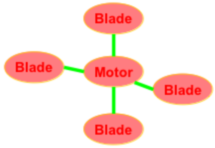
- semantic understanding, annotation, and recognition;
- computation of $\mathbb{Z}$-homology.



Topological data structure (IA$*$)



Structural model (MC-decomposition)



Semantic model (future work)

Acknowledgments

We thank:

- the Italian Ministry of Education and Research (the PRIN 2009 program);
- the National Science Foundation (contract IIS-1116747);
- Dott. Daniele Panozzo (DISI, Genova) for *CCViewer* and useful suggestions.

These slides wiil be available on
`http://www.disi.unige.it/person/CaninoD`.

Thanks for your attention!

References

D. Boltcheva, D. Canino, S. Merino, J.-C. Leon, L. De Floriani, F. Hetroy *An iterative algorithm for homology computation on simplicial shapes*, Computer-Aided Design - Proc. of GD/SPM 2011, vol. 43, pp.1457-1467

D. Canino, L. De Floriani, K. Weiss, *IA^* : an adjacency-based representation for non-manifold simplicial shapes in arbitrary dimensions*, Computer & Graphics - Proc. of SMI Conf. 2011, vol. 35, issue 3, pp. 747-753

L. De Floriani L., A. Hui, D. Panozzo, D. Canino, *A dimension-independent data structure for simplicial complexes*. In Proc. of International Meshing Roundtable 2010, pp. 403-420

Edelsbrunner H., *Algorithms in Combinatorial Geometry. Springer, 1987*