Combinatorial 3-manifolds from sets of tetrahedra

Marco Attene
IMATI-GE
CNR
Italy
marco.attene@ge.imati.cnr.it

Massimo Ferri
Dept. of Mathematics
University of Bologna
Italy
ferri@dm.unibo.it

Daniela Giorgi
IMATI-GE
CNR
Italy
daniela.giorgi@ge.imati.cnr.it

Abstract

We propose an algorithm to convert a tetrahedral mesh with singularities to a combinatorial 3-manifold using only local modifications. We outline sufficient conditions on the mesh to guarantee the feasibility of the approach and we show how singularities can be both identified and removed according to the configuration of their link. Furthermore, we demonstrate that the algorithm can be implemented using a flexible state-of-the-art data structure for manifold tetrahedral meshes suitable for efficient and general applications.

1 Introduction

Virtual 3D shapes are crucial in many sectors such as industrial design, gaming, simulation and medicine, to cite a few, and can be either conceived using computer-aided tools, or reconstructed out of digitized real 3D objects. 3D shapes can be modelled in several ways [23], though the most common approach relies on the so-called *Boundary Representation* (B-Rep), in which a solid is represented indirectly as the volume bounded by a given, explicit surface. B-Reps are particularly appropriate for both the designer and the computer; NURBS, for example, make it possible for the designer to control the shape of smooth surface patches through few control points, while, from the point of view of the computer, triangle meshes are directly supported by the graphic hardware for rendering complex shapes at exceptional speeds.

Nevertheless, in some cases it is necessary to explicitly model also the inner parts of the shape. Fully solid shape models, for example, come directly from CT or MRI scans in medicine. In this case, the volume is typically represented in raster form as the space being digitized is subdivided in a regular grid of voxels.

In other scenarios, a B-Rep may need to be converted to a volumetric simplicial mesh (i.e. a tetrahedral mesh) in order to apply physically-based simulation techniques. In Computer Graphics, for example, realistic simulation of deformable objects are based on volume meshes [26] while, more in general, in computational sciences numerical solvers for partial differential equations need a discrete domain to apply finite-element or finite-volume methods.

The representation of simplicial meshes in a computer has been widely studied, and a number of data structures have been proposed in the literature [3, 5, 6, 9]. The most efficient structures, however, can represent only manifold meshes, hence a lot of research has been dedicated to the conversion of generic simplicial meshes to *more efficient* manifold meshes. Unfortunately, while state-of-the-art solutions to convert surface meshes are satisfactory, for higher-dimensions some more problems have been encountered [10].

1.1 Motivation

In medical applications, CT or MRI scans of the patient generate raster 3D images in which the brightness of each voxel is related to the type of tissue sampled in that position. In several cases it is important to *extract* from the 3D image the shape of a particular tissue (e.g. an organ or a bone). This procedure, known as *segmentation*, may generate volumetric simplicial meshes [2, 17], and their suitability for specific kinds of analysis often requires them to be manifold [11].

Similarly, recent variational meshing techniques [1] allow one to easily convert a surface mesh to a volume mesh in which the placement of vertices in inner parts guarantees both a smooth transition in sampling density and well-shaped tetrahedra (these two conditions are fundamental for an effective application of the finite-element method). Nevertheless, such volumetric models are not guaranteed to be manifold (see Figure 1), and this is a limitation in several other applications where efficiency is mandatory.

On a data structure designed for manifold meshes specifically, in fact, traversal operations are much faster, and this

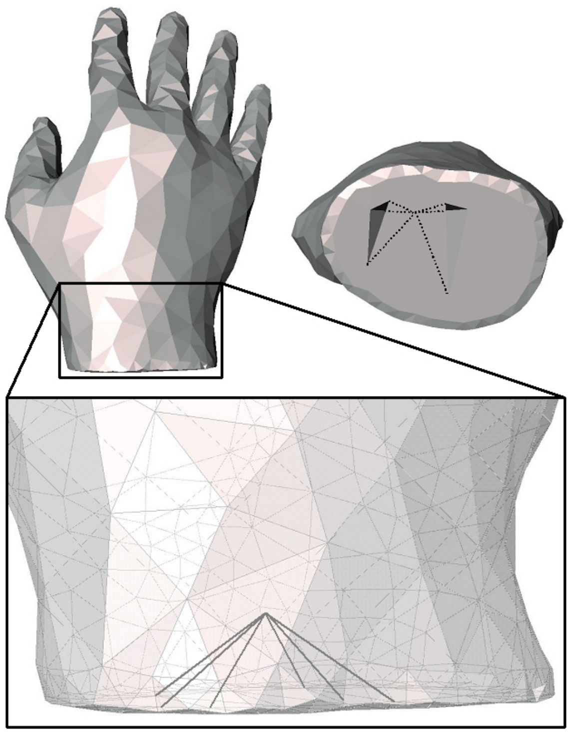

Figure 1. A tetrahedral mesh obtained through the variational meshing algorithm described in [1] in which a singular vertex is shown. Model courtesy of Pierre Alliez.

is fundamental to implement efficient algorithms to detect collisions, to perform boolean or morphing operations, to simplify the model, or even simply to render it efficiently.

Note that most data structures neatly separate the connectivity of the mesh, which is an abstract simplicial complex, from its geometry, which defines the position of each vertex in the 3D space. Thus, since typical traversal operations are independent of the geometry, efficiency is granted by the *manifold* condition on the abstract complex defining the connectivity.

Hence, it is important to develop efficient approaches to convert the connectivity of a volumetric simplicial mesh into a manifold abstract complex.

1.2 Overview and contributions

In this paper an algorithm is proposed to process the connectivity of a tetrahedral mesh so as to make it manifold. Existing related work is described in section 2, where both the most important state-of-the-art algorithms used to process B-reps and the attempts done so far to treat higher dimensional complexes are presented. The mathematical background is then presented in section 3 and, based on this, the problem tackled in the paper is stated formally. The approach is then described in section 4 both at a conceptual level using a mathematical terminology, and at a practical level using a more computer-driven language and pseudocode snippets. In section 5 we discuss some of the choices made in the design of the algorithm and outline possible alternative approaches. Finally, we draw our conclusions and the directions of future research in section 6.

2 Related Work

Representations for non-manifold objects exist and are useful in several contexts [16, 19, 22]. Nevertheless, the need for efficient algorithms has called for the development of data structures specifically dedicated to the manifold case. Manifold simplicial complexes can be represented in any dimension [4, 24], although most data structures have been proposed specifically for the 2D and 3D cases [5, 7, 15, 21, 23]. In several practical applications, it happens that the simplicial complex resulting from a specific process (e.g. segmentation of a 3D image or digitization of a real object) is mostly manifold, in the sense that only a small percentage of vertices are singular. This fact prompted the development of several algorithms that slightly modify the complex to remove the singularities without changing anything far from them. In this context, a widely used approach consists of decomposing non-manifold complexes into simpler parts, splitting at those elements (vertices, edges, facets, etc.) where singularities occur [8, 14, 25]. The result of such a decomposition is a collection of singularity-free components that can be represented by standard data structures for manifold complexes.

2.1 Manifold surface meshes

Most of these works have been proposed for the case of surface meshes. In [25], for example, a method is proposed to convert a non-manifold set of triangles to a set of manifold surface meshes by first identifying non-manifold edges (i.e. edges having more than two incident faces); within the data structure, each such edge is replicated a number of times sufficient to assign at most two incident faces to each copy. In a second step, non-manifold vertices are also identified and duplicated properly.

In a similar setting, [14] identifies non-manifold edges and cuts the surface along such edges, that is, each non-manifold edge having k incident faces is turned into k boundary edges having 1 incident face each. In a second step, the user may choose whether to *pinch* or to *snap* such boundary edges; in the former case, each boundary loop is simply zipped, or closed, while in the latter case pairs of neighboring boundary loops are merged together.

2.2 Manifold complexes in higher dimensions

In [10], it has been pointed out that the decomposition of a non-manifold complex should not introduce artificial or arbitrary *modifications* to manifold parts. Under these assumptions, a decomposition into manifold components is possible, in general, only for two-dimensional complexes. Therefore, instead of trying to obtain a set of manifold components, the solution proposed in [10] converts a general non-manifold complex to a set of so-called *initial quasi-manifolds*. These objects are equivalent to actual manifolds in dimension 2, while in higher dimensions they may contain singular vertices (Figure 2). Thus, to efficiently work with initial quasi manifolds, proper data structures are required in which the presence of such singularities do not represent a drawback; in [18] such a data structure has been proposed for the specific 3D case. Though initial quasi-manifolds have several interesting characteristics, and though they can be represented through proper data structures, the presence of singularities make them still not satisfactory for a number of applications.

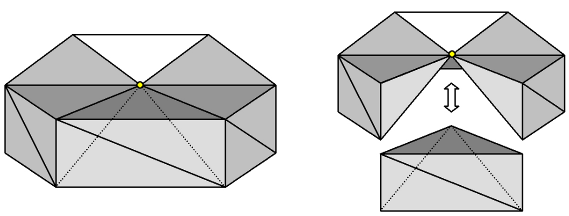

Figure 2. An example singular vertex in a 3D initial quasi-manifold. On the right, a pair of adjacent tets have been detached to better show the singularity. In [10], this configuration is called a *pinched pie***.**

3 Background definitions

3.1 Simplicial complexes

A k-$dimensional$ $simplex$, or k-$simplex$, A^k is a set $V = \{v_0, \ldots, v_k\}$ of $k+1$ objects called *vertices*, together with the set of real-valued functions $\alpha : V \to \mathbb{R}$ satisfying $\sum_{v_i \in V} \alpha(v_i) = 1$ and $\alpha(v_i) \geq 0$. A function α is called a point of A^k. The values $\alpha(v_0), \ldots, \alpha(v_k)$ are the *barycentric coordinates* of the point α [12].

A *geometric realization* $|A^k|$ of A^k in the Euclidean space $\mathbb{R}^n$, $n \geq k$, can be obtained by defining a bijection between the vertices of A^k and a set of $k+1$ affinely independent points $p_0, p_1, \ldots, p_k$ of $\mathbb{R}^n$, so that $|A^k| = \{(t_0p_0 + t_1p_1 + \ldots + t_kp_k) \in \mathbb{R}^n \mid t_i \geq 0, \sum_i t_i = 1\}$. Thus, $|A^k|$ is the convex hull of $p_0, \ldots, p_k$. In particular, the *standard k-simplex* Δ^k is defined as the convex hull of the points $e_0 = (1, 0, \ldots, 0)$, $e_1 = (0, 1, 0, \ldots, 0)$, ..., $e_k = (0, 0, \ldots, 1) \in \mathbb{R}^{k+1}$.

A (proper) *face* B of A^k, denoted $B < A^k$, is a simplex determined by the (proper) subset $W \subset V$, whose points $\beta : W \to \mathbb{R}$ are identified with the points $\alpha : V \to \mathbb{R}$ such that $\alpha(v_i) = \beta(v_i)$ if $v_i \in W$ and $\alpha(v_j) = 0$ if $v_j \in V - W$. If B is a face of A, then A is said to be *incident* at B.

A *finite simplicial complex* K is a finite set of simplices such that:

i) if $A \in K$ and $B < A$, then $B \in K$;

ii) if $A, B \in K$, then $A \cap B$ is either empty or it is a face of both A and B.

From now on, we shall omit the term finite.

The *underlying space* $|K|$ of K is the union $\cup_{A \in K} |A|$ of the geometric realization of its simplices.

The *dimension* of K is the dimension of the largest dimensional simplex belonging to K. A simplicial complex of dimension n is *homogeneous* if it is made of n-simplices and their faces.

L is a *subcomplex* of K if L is a complex and $L \subset K$. For $A \in K$, the (closed) *star* of A in K, $star(A, K)$, is the subcomplex of K made of all simplices of K having A as a face plus all their faces. If $A \in K$, then the *link* of A in K, $link(A, K)$, is the set of simplices in $star(A, K)$ whose intersection with A is empty.

The *boundary* ∂A of a simplex A is the complex made of the proper faces of A. The *boundary* ∂K of a homogeneous n-dimensional simplicial complex K is the $(n-1)$-complex obtained as the sum mod 2 of the $(n-1)$-dimensional simplices of the boundary ∂A of each of the n-simplices $A \in K$ plus their faces [12].

3.2 Piecewise linear homeomorphisms

Let K and L be simplicial complexes whose sets of vertices are denoted $V(K)$ and $V(L)$, respectively, and let $f : V(K) \to V(L)$ be a bijection such that the vertices $v_0, \ldots, v_p$ of K span a simplex of K if and only if $f(v_0), \ldots, f(v_p)$ span a simplex of L. Then, f is said to be a *simplicial isomorphism* and the induced map $|f| : |K| \to |L|$, taking $v = \sum_i t_i v_i$ to $g(v) = \sum_i t_i f(v_i)$, is called a *piecewise linear homeomorphism*.

A complex L is a *subdivision* of the complex K if $|L| = |K|$ and every simplex of L lies in a simplex of K. A map $g : |K_1| \to |K_2|$ is then a piecewise linear homeomorphism if and only if there exist subdivisions L_1 of K_1 and L_2 of K_2 and a simplicial isomorphism $f : L_1 \to L_2$ such that $g = |f|$. $|K_1|$ and $|K_2|$ are said to be *piecewise linear homeomorphic* if there exists a piecewise linear homeomorphism between them. For the sake of simplicity we shall confuse, with a fairly usual abuse of language, every K with $|K|$, f with $|f|$, and the notion of simplicial isomorphism with that of piecewise linear homeomorphism.

A *combinatorial n-ball* is a complex piecewise linearly homeomorphic with the standard simplex Δ^n. A *combinatorial n-sphere* is a complex piecewise linearly homeomorphic with the boundary $\partial\Delta^{n+1}$ of the standard simplex Δ^{n+1}.

3.3 Combinatorial n-manifolds

A *combinatorial n-manifold* is a homogeneous n-dimensional complex K such that for any vertex v of K, $link(v, K)$ is a combinatorial $(n-1)$-ball if $v \in \partial K$ and a combinatorial $(n-1)$-sphere if $v \notin \partial K$.

A simplex $A^p \in K$ is *regular* in K, where K is a homogeneous n-dimensional complex, if $link(A^p, K)$ is a combinatorial $(n-p-1)$-ball if $A^p \in \partial K$ and a combinatorial $(n-p-1)$-sphere if $A^p \notin \partial K$; otherwise A^p is called a $singular$ simplex. It follows that a combinatorial n-manifold is a homogeneous complex in which every vertex is regular. It also holds that in a combinatorial n-manifold all simplices are regular [12].

4 Building combinatorial 3-manifolds

In this paper, we deal with three-dimensional simplicial complexes. In this case, special names are given to simplices depending on their dimension. Specifically, a 3-simplex is called a $tetrahedron$, or simply tet, a 2-simplex is a $facet$, a 1-simplex is an $edge$ and a 0-simplex is a $vertex$. We adopt the term *tetrahedrization* to indicate a complex made of tetrahedra and their faces.

4.1 Problem statement

Le K be a homogeneous, possibly non manifold subcomplex of a combinatorial 3-ball; without loss of generality, we assume that K has no vertices on the boundary of the ball. Our aim is to locally edit K so as to transform it in a manifold complex. Editing operations must influence the neighborhood of the singularities exclusively, while the remaining parts of the complex must remain unmodified. More in detail, supposing, without loss of generality, that K has a single singular vertex (or edge) v, we shall obtain a new complex K' that will be a combinatorial 3-manifold coincident with K everywhere but $star(v, K)$. Note that, differently from [10], we allow the modification of some manifold faces which are incident at singular elements.

It is possible to prove that the new complex K' can be geometrically realized in the Euclidean space, so that $|K| \subset |K'|$ or $|K'| \subset |K|$. Since the aim of this paper is to fix the connectivity of the complex, we will not investigate into these issues. We will limit ourselves to showing a pseudo-realization of K' in $|K|$, that is we will indicate a non-injective simplicial map from K' to K.

4.2 Approach

Roughly speaking, our approach consists of two phases: first we identify and treat singular edges; in a second step, we deal with the remaining vertex singularities adopting two different procedures that depend on the configuration of the link.

More formally, let L be a combinatorial 3-ball, and K a homogeneous sub-complex of L with no vertices in ∂L. The procedure we adopt to remove singularities located at vertices requires that the link of each vertex has no singularities. To achieve this condition, we treat singular edges first as follows (refer to Figure 3 for an example showing a step of the procedure).

Let $e = \{v_1, v_2\}$ be a singular edge in K. Then $link(e, K)$ is a simplicial 1-complex made of $k > 1$ components $L_i(e)$, with $i = 1, \ldots, k$. We create k new vertices w_i, with $i = 1, \ldots, k$, each positioned at the midpoint of e. For each such vertex, we also create two new edges $e_1^i = \{v_1, w_i\}$ and $e_2^i = \{w_i, v_2\}$. Now, for each edge $l^{i,j} = \{u_1^{i,j}, u_2^{i,j}\}$ in $L_i(e)$, with $i = 1, \ldots, k$, we consider the tet $t^{i,j}$ having both $l^{i,j}$ and e as faces, and replace it with two new tets $t_1^{i,j} = \{v_1, w_i, u_1^{i,j}, u_2^{i,j}\}$ and $t_2^{i,j} = \{w_i, v_2, u_1^{i,j}, u_2^{i,j}\}$. After these operations, the original singular edge e has no remaining incident tets and is removed from the complex.

In the new complex obtained by applying this procedure to all the singular edges, that we still call K, all the newly created w_is are regular and all the singular vertices have a manifold link.

Each remaining singular vertex v in K can then be treated as follows.

By hypothesis, $link(v, L)$ is a combinatorial 2-sphere while $link(v, K)$ is neither a combinatorial 2-sphere, nor a combinatorial 2-ball. Since $link(v, K)$ is a sub-complex of $link(v, L)$, $link(v, K)$ is the disjoint union of $n \geq 1$ combinatorial 2-spheres with holes. The boundary $\partial link(v, K)$ of $link(v, K)$ is made up of $k > 1$ components, each of which is a combinatorial 1-sphere.

We proceed recursively with respect to k.

Among the combinatorial 1-spheres bounding $link(v, K)$, at least one of them, let it be C, bounds a 2-ball D of either $link(v, K)$ or $link(v, L - K)$ (note that in our setting the Schoenflies conjecture holds [13]).

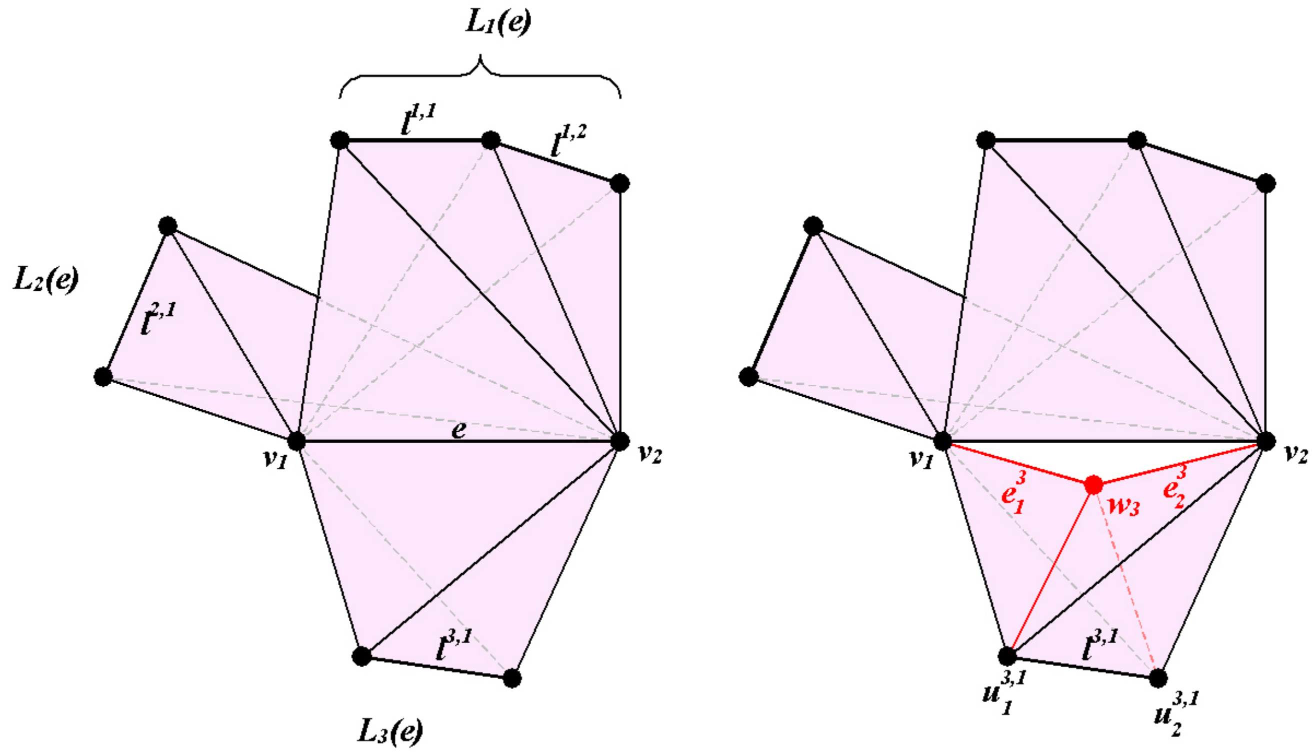

Figure 3. An example of singular edge e in which one of the components ($L_3(e)$) of its link is considered for the creation of a new vertex w_3 and its incident elements. The algorithm places the new vertex w_3 at the midpoint of the singular edge e; in the figure, however, it is depicted in a displaced position to better illustrate the resulting connectivity.

We add a new vertex w in K at the same position as v, and distinguish two possible cases:

i. If $D \subset link(v, K)$, we replace each tet t_i incident at v and having a face in D with a tet τ_i obtained from t_i by substituting v with w (Figure 4).

ii. If $D \subset link(v, L - K)$, we consider the facets f_i incident at v and having an edge in C and, for each of them, we add a new tet τ_i made of w and the vertices of f_i (Figure 5).

Now, the new vertex w is clearly regular, while the boundary $\partial link(v, K)$ is made of $k - 1$ components, each of which is a combinatorial 1-sphere. Hence, we repeat the procedure as long as the number of components in $\partial link(v, K)$ is greater than 1.

In this way, we obtain a new complex K' in which all simplices are regular, that is a combinatorial 3-manifold.

Notice that, for simplicity, the new vertices w are assigned the same coordinates of the singular vertex v in the Euclidean space. This implies that our method does not produce a geometric realization of K in the Euclidean space.

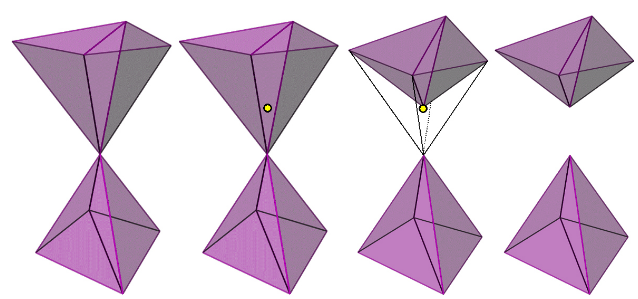

Figure 4. Vertex duplication and retriangulation of a configuration having two connected components in the link of an isolated singular vertex. The algorithm places the new vertex at the same position of the former singularity; in the figure, however, it is depicted in a displaced position to better illustrate the resulting connectivity.

Actually, what happens is that a map $f : V(K') \rightarrow V(K)$ induces a simplicial map which is not injective.

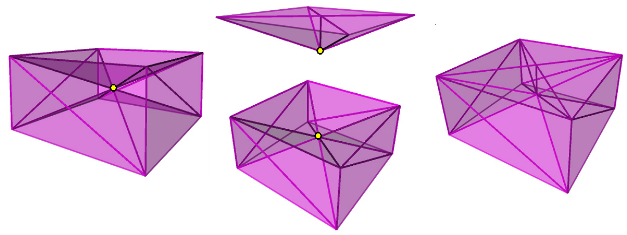

Figure 5. Conversion of a *pinched pie* **configuration to a combinatorial 3-ball. The algorithm places the new vertex at the same position of the former singularity; in the figure, however, it is depicted in a displaced position to better illustrate the resulting connectivity.**

4.3 Algorithm

We have implemented the algorithm based on the *tetrahedral data structure* introduced in [5]. Within such a structure, the four basic entities of a tetrahedrization (i.e. vertices, edges, facets and tets) are encoded, while only a minimal set of topological relations are stored explicitly. Topological relations describe the connectivity of the complex by linking each simplex to its *boundary* and *co-boundary*, that is, the set of its faces and the set of its incident simplices respectively.

For compactness, the tetrahedral data structure explicitly encodes only the following four *constant* relations, each of which links an element with a constant number of neighboring entities:

- **tet-facet** (TF) which returns the four facets bounding a tetrahedron;
- **facet-tet** (FT) which returns the tets (one or two) incident at a facet;
- **facet-edge** (FE) which returns the three edges bounding a facet;
- **edge-vertex** (EV) which returns the two vertices of an edge;

and the following two special relations, each of which links an element with only one of the neighboring entities:

- **edge-facet** (EF*) which returns one of the facets incident at an edge;
- **vertex-edge** (VE*) which returns one of the edges incident at a vertex.

Provided that the tetrahedrization encoded is combinatorially manifold, from this minimal set it is possible to compute all the other topological relations in optimal time (i.e. in a number of operations linearly proportional to the number of elements involved in the relation being computed). We point the reader to [5] for a detailed description of the algorithms that compute these *implicit* relations.

Note that, if the tetrahedrization *T* is a sub-complex of a combinatorial ball, each facet can have at least one, and at most two incident tets; in this case, even if *T* is not a combinatorial manifold, it can be encoded by the tetrahedral data structure and all the constant relations can be still computed in optimal time.

The presence of singularities, however, does not allow to extract all the other relations in optimal time. Therefore, to keep the complexity of the conversion algorithm linear, we pre-compute the link of all the vertices and edges as follows.

Each vertex *v* is assigned an additional relation *L(v)*, initially empty, that encodes its link. Note that, since such a link is a homogeneous simplicial 2-complex, L needs to explicitly store only the facets; all the other lower-dimensional elements of the link are faces of such facets and can be extracted through the FE and EV relations. Then, for each tet *t* in the tetrahedrization we consider its four facets, and add each such facet to the link L of its opposite vertex in *t*. This procedure is sketched in the pseudo-code of Listing 1, where the constant relation FV returning the vertices of a facet is obtained as the combination $FV(f) = EV(FE(f))$, and the constant relation TV returning the vertices of a tet is obtained as the combination $TV(t) = FV(TF(t))$.

The star of *v* can be computed starting from *L(v)* in optimal time through the FT relation.

Listing 1. Construction of the links of all the vertices

```
CreateVertexLinks(Tetrahedrization T)
{
  for each vertex v in T
   L(v) := new empty facet list

  for each tet t in T
   for each facet f in TF(t)
   {
    v := the vertex of TV(t) which is not in FV(f)
    Add f to L(v)
   }
}
```

Similarly, each edge *e* is assigned an additional relation *L(e)* that encodes the edges constituting its link, which is a homogeneous simplicial 1-complex. Thus, for each tet *t* in the tetrahedrization we consider its six edges, and add each such edge to the link *L(e)* of its opposite edge *e* in *t*.

Unfortunately, the star of *e* cannot be computed starting from *L(e)* in optimal time. Hence, during the computation of *L(e)*, we also fill a further relation *T(e)* with the tets t used to build *L(e)* (i.e. the tets incedent at e that, together with their faces, constitute the star of e).

The procedure to compute the links and the stars of all the edges is sketched in Listing 2, where the relation TE is computed as the composition of FE and TF.

Listing 2. Construction of the links and the stars of all the edges

```
CreateEdgeLinksAndStars(Tetrahedrization T)
{
  for each edge e in T
  {
   L(e) := new empty edge list()
   T(e) := new empty tet list()
  }

  for each tet t in T
   for each edge e in TE(t)
   {
    e2 := the edge in TE(t) sharing no vertex with e
    Add e to L(e2) and t to T(e2)
   }
}
```

Afterwards, the link *L(e)* of each edge is analyzed, and if it is not connected *e* is declared to be a singular edge.

In this case, one component of the link remains as it is, while for each additional component $L_i(e)$ of *L(e)* a copy e_i of e is created, and the following topological relations are updated in the data structure:

- EV(e_i) := EV(e)
- for each tet t in $T(e)$ having an edge in $L_i(e)$, and for each facet f of t such that FE(f) contains e, replace e with e_i in FE(f). Take one of these facets as the value of EF*(e_i)

Finally, e and the e_is are split at their midpoints to eliminate duplicated edges in the structure, and the L and T structures are updated accordingly for all the elements involved in the modification.

At this stage, we are guaranteed that the link of each vertex is a combinatorial 2-manifold with some boundaries and possibly more than one connected component.

In particular, if the link *L(v)* of *v* has more than one boundary component, then *v* is declared to be singular. In this case, the algorithm proceeds as described in Listing 3.

Once such algorithm terminates, the data structure represents a combinatorial 3-manifold, thus the L and T relations can be deleted, and all the topological relations can now be extracted in optimal time.

Since the tetrahedral data structure is suitable for a wide spectrum of applications, it is worth to implement the conversion algorithm directly on such a structure. Note that the use of this structure has no effects on the overall complexity of the algorithm; all the operations needed to perform the conversion, in fact, can be executed in optimal time.

5 Discussion

Our method assumes that the input mesh is a homogeneous subcomplex of a combinatorial ball. Note that this condition is not too restrictive. In medicine, for example, segmented 3D images can generate tetrahedral meshes by simply triangulating the voxels; though not necessarily manifold, the resulting mesh is clearly a subcomplex of a combinatorial ball (i.e. the fully tetrahedrized 3D image). Also, our algorithm is applicable to the tetrahedrizations produced by popular meshing methods (e.g. [1]) based on a constrained Delaunay tetrahedrization of the input's convex hull.

The approach we have chosen is inspired by existing methods for surface meshes. In the case of surfaces, however, all the isolated singular vertices can be treated using the same scheme [25]. Conversely, for tetrahedral meshes we need to use different procedures depending on the configuration of the link of each singularity, and each such procedure has a different impact on the type of connectivity of the mesh.

Specifically, to treat a singular vertex we need to analyze its link and recursively apply a proper procedure to each component of the link. If the component is a combinatorial 2-ball we duplicate the singular vertex by changing the type of connectivity of its star (i.e. its fundamental group); otherwise, if the component is a 2-ball with n holes, we $emboss$ the singularity n times without modifying the fundamental group of its star. These two strategies, namely $i.$ and $ii.$ in section 4.2, are illustrated in Figure 4 and Figure 5 respectively.

Though the modification of the fundamental group deriving from our approach is suitable for most real-world practical applications, alternative solutions can be implemented. For example, the two combinatorial cones depicted in Figure 4 can be merged together instead of being separated; similarly, the pinched pie configuration in Figure 5 can be converted to a solid torus instead of becoming a ball. The difference between the results of these approaches can be seen in the real-world example illustrated in Figure 6.

Note that the same arguments have been dealt with for the case of surface meshes in [14], where surface patches connected through singular points can be either separated or $regularly$ merged.

Listing 3. Conversion algorithm

```
EditSingularVertex(Vertex v)
{
 1.  Look for a connected component in L(v) which is a combinatorial 2-ball
 2.  if there is such a component, let it be D
 3.    Compute the set T of tets incident at v and having a facet in D
 4.    Create a new vertex w
 5.    Remove all the tets in T from the tetrahedrization
 6.    Create a new tet {w,w1^i,w2^i,w3^i} for each facet {w1^i,w2^i,w3^i} in D
 7.    Remove D from L(v)
 8.    If L(v) has more than one component go to 1, else go to 15
 9.  else (i.e. if there are no combinatorial 2-balls in L(v))
 10.   Extract a boundary loop B of L(v)
 11.   Compute the set D of facets incident at v and having an edge in B
 12.   Create a new vertex w
 13.   Create a new tet {w,w1^i,w2^i,v} for each facet {w1^i,w2^i,v} in D
 14.   Add all the facets {w,w1^i,w2^i} to L(v)
 15.   If L(v) has more than one boundary component go to 10, else terminate
}
```

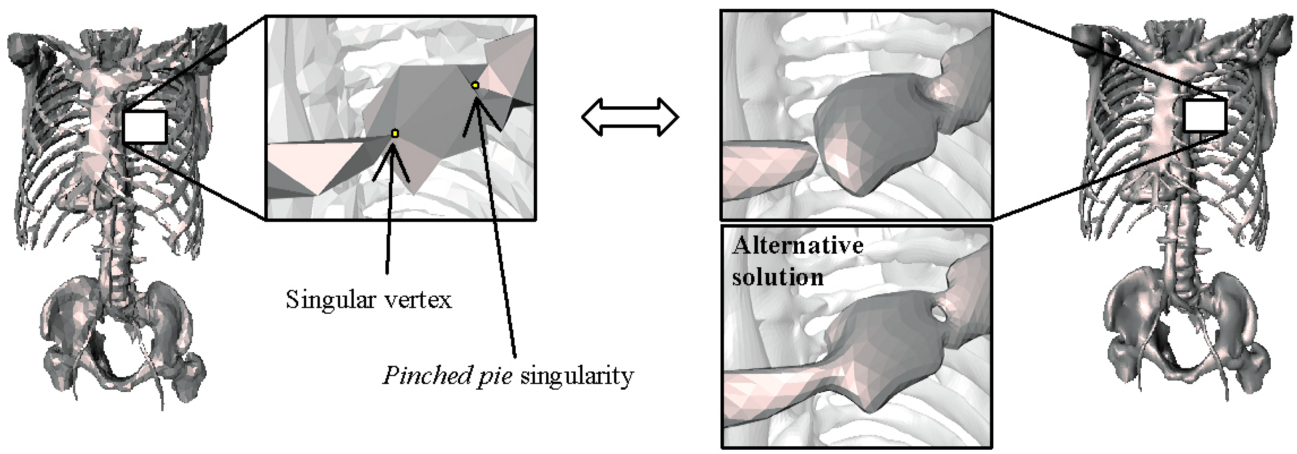

Figure 6. Through our approach, a tetrahedral mesh resulting from a segmentation of a medical image (left) is converted to a combinatorial manifold. Its boundary has been subdivided twice using Loop's scheme [20] (right) to show the resulting connectivity around formerly singular vertices (magnified). An additional magnification shows the result of the same part subdivided after being processed by the alternative approaches discussed in section 5.

6 Conclusions and Future Research

We have shown that, under specific conditions, a tetrahedral mesh with singularities can be converted to a combinatorial 3-manifold by using only local modifications. Based on well-established mathematical concepts, we have outlined sufficient conditions that make such a conversion possible, and developed a novel conversion algorithm. Moreover, we have shown that the algorithm can be implemented on a data structure designed for manifold complexes; in this way, the algorithm can be implemented as part of more general applications dealing with manifold tet meshes without the need of introducing additional structures.

In future work, we wish to extend the approach so that, besides reaching the manifoldness in a combinatorial sense, it also guarantees the manifoldness of the underlying space of the complex.

Acknowledgement

This work has been supported by CNR research activity ICT-P03 and the European Network of Excellence FP6 IST

NoE 506766 “AIM@SHAPE”, and performed within the activity of ARCES “E. De Castro”, University of Bologna, under the auspices of INdAM-GNSAGA. The authors thank B. Falcidieno and M. Spagnuolo for fruitful discussions.

References

[1] P. Alliez, D. Cohen-Steiner, M. Yvinec, and M. Desbrun. Variational tetrahedral meshing. *ACM Transactions on Graphics*, 24(3):617–625, 2005.

[2] S. Battiato, C. Bosco, G. Farinella, and G. Impoco. 3D CT segmentation for clinical evaluation of knee prosthesis operations. In *Procs of 4th Eurographics Conference - Italian Chapter*, Feb 2006.

[3] B. G. Baumgart. A polyhedron representation for computer vision. In *Procs of National Computer Conference*, pages 589–596, 1975.

[4] E. Brisson. Representing geometric structures in d dimensions: topology and order. In *Procs of 5th ACM Symp. on Computational Geometry*, pages 218–227, New York, Jun 1989. ACM Press.

[5] E. Bruzzone and L. D. Floriani. Two data structures for building tetrahedralizations. *The Visual Computer*, 6(5):266–283, 1990.

[6] S. Campagna, L. Kobbelt, and H. P. Seidel. Directed edges - a scalable representation for triangle meshes. *ACM Journal of Graphics Tools*, 3(4), 1998.

[7] D. Dobkin and M. Laszlo. Primitives for the manipulation of three-dimensional subdivisions. *Algorithmica*, 5(4):3–32, 1989.

[8] B. Falcidieno and O. Ratto. 2-manifold cell decomposition of r-sets. *Computer Graphics Forum*, 11(3):391–404, 1992.

[9] L. D. Floriani and A. Hui. Data structures for simplicial complexes: an analysis and a comparison. In *Third Eurographics Symposium on Geometry Processing*. Eurographics Association, jul 2005.

[10] L. D. Floriani, M. M. Mesmoudi, F. Morando, and E. Puppo. Decomposing non-manifold objects in arbitrary dimensions. *Graphical Models*, 65:2–22, 2003.

[11] C. Forest, H. Delingette, and N. Ayache. Removing tetrahedra from manifold tetrahedralisation: application to real-time surgical simulation. *Madical Image Analysis*, 9:113–122, 2005.

[12] L. C. Glaser. *Geometrical Combinatorial Topology*, volume 1. Van Nostrand Reinhold, 450 West 33rd Street, New York, 1970.

[13] L. C. Glaser. A proof of the most general polyhedral schoenflies conjecture possible. *Pacific Journal of Mathematics*, 38(2):401–417, 1971.

[14] A. Gueziec, G. Taubin, F. Lazarus, and B. Horn. Cutting and stitching: Converting sets of polygons to manifold surfaces. *IEEE Trans. on Visualization and Computer Graphics*, 7(2):136–151, 2001.

[15] L. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and computation of voronoi diagrams. *ACM Transactions on Graphics*, 4(2):74–123, 1985.

[16] E. Gursoz, Y. Choi, and F. Prinz. Vertex-based representation of non-manifold boundaries. In M. J. Wozny, J. Turner, and K. Preiss, editors, *Geometric Modeling for Product Engineering*, pages 107–130. Elsevier, North Holland, 1990.

[17] U. Hartmann and F. Kruggel. A fast algorithm for generating large tetrahedral 3D finite element meshes from magnetic resonance tomograms. In *Procs. of the IEEE Workshop on Biomedical Image Analysis*, page 184, 1998.

[18] A. Hui, L. Vaclavik, and L. D. Floriani. A decomposition-based representation for 3D simplicial complexes. In *Fourth Eurographics Symposium on Geometry Processing*. Eurographics Association, jul 2006.

[19] S. Lee and K. Lee. Partial entity structure: a fast and compact non-manifold boundary representation based on partial topological entities. In *Procs of 6th ACM Symp. on Solid Modeling and Applications*, pages 159–170, New York, 2001. ACM.

[20] C. Loop. Smooth subdivision surfaces based on triangles. *Ms.C Thesis, University of Utah, USA*, 1987.

[21] H. Lopes and G. Tavares. Structural operators for modeling 3-manifolds. In *Procs of 4th ACM Symp. on Solid Modeling and Applications*, pages 45–60, New York, May 1997. ACM.

[22] Y. Luo and G. Lukacs. A boundary representation of form features and non-manifold solid objects. In *Procs of 1st ACM Symp. on Solid Modeling and Applications*, pages 45–60, New York, Jun 1991. ACM.

[23] M. Mäntylä. *Introduction to Solid Modeling*. Computer Science Press, Rockville, Maryland, USA, 1988.

[24] A. Paoluzzi, F. Bernardini, C. Cattani, and V. Ferrucci. Dimension-independent modeling with simplicial complexes. *ACM Transactions on Graphics*, 12(1):56–102, 1993.

[25] J. Rossignac and D. Cardoze. Matchmaker: manifold breps for non-manifold r-sets. In *Procs of 5th ACM symposium on Solid modeling and applications*, pages 31–41, New York, NY, USA, 1999. ACM Press.

[26] Y. Zhuang and J. Canny. Real-time and physically realistic simulation of global deformation. In *SIGGRAPH99 Sketches and Applications*, New York, NY, USA, aug 1999. ACM Press.