STAG: Smart Tools and Apps in computer Graphics (2016)
Giovanni Pintore and Filippo Stanco (Editors)

Manipulating Topological Decompositions of Non-Manifold Shapes

David Canino, Independent Researcher, PhD. in Computer Science, canino.david@gmail.com

Abstract

Decomposing a non-manifold shape into its almost manifold components is a powerful tool for analyzing its complex structure. Many techniques for decomposing a non-manifold shape are available in the current literature, and provide a structural model*, which exposes its non-manifold singularities, as well as the connectivity of its relevant subcomponents, connected through the singularities. However, the majority of the decompositions are static, and are not automatically updated, if the corresponding non-manifold shape is modified by an editing operator. In many cases, the resulting decomposition is recomputed from scratch without reusing the unchanged portions of the existing decomposition. In this paper, we describe how updating automatically a specific decomposition of a non-manifold shape. Here, we show that our approach may be useful for adapting many geometry processing techniques also to non-manifold shapes, where several problems may arise. One of the most promising applications consists of defining a multiresolution version for the specific structural model of interest, due to its* good *topological properties.*

1. Introduction

Decompositions are a powerful tool for modeling the *non-manifold shapes*. Informally, a non-manifold shape is a subset of the Euclidean space, such that every neighborhood of at least one point p is not locally homeomorphic to a sphere, centered at p. The point p with these properties is a *non-manifold singularity*. These shapes are common in the engineering applications [BCMA*11], but their efficient manipulation is challenging. Many geometry processing applications are defined on the manifolds [BKP*10], and their extension to the non-manifolds is not trivial.

The digital shapes are often discretized by the *cell complexes*, and the *topological data structures* are the most common tools for representing the local connectivity of the elementary cells in the complexes [DFH05, BKP*10]. However, many of these representations do not describe the *structure* of the non-manifold shapes, and do not expose the non-manifold singularities, except some exceptions [Can12]. This is an important aspect, since a dimension-independent algorithm for recognizing the manifold shapes does not exist, thus this operation is not always *computable* [Nab96].

In order to overcome these limitations, a non-manifold shape is decomposed into the collection of its *meaningful* components, that are easily distinguished from the remaining portions of the shape. The most natural decomposition consists of cutting a non-manifold shape along its non-manifold singularities without modifying its almost manifold parts. The resulting representation highlights the subcomponents and their connectivity, and the non-manifold singularities are exposed explicitly. Broadly speaking, this representation provides the *structural model* (or the *decomposition*) of a non-manifold shape, and exposes its high-level description in terms of the subcomponents. Here, several details are abstracted, and the geometric, the combinatorial, and the semantic aspects are decoupled in order to be integrated more easily and efficiently.

1.1. Prior and Related Work

In the current literature, many structural models are defined for the manifold shapes, and are called the *segmentations* [Sha08]. A manifold shape is decomposed into the collection of its meaningful components, called the *segments*, that are defined by several metrics and application-dependent criteria.

Our objective consists of cutting a non-manifold shape into the collection of its manifold components along the non-manifold singularities without modifying arbitrarily the components. This implies that the structural model of a manifold shape coincides with the shape itself, since there is no singularity. Instead, a shape is decomposed into several segments. In any case, it is also possible to compute the segmentation of a component in the structural model of a non-manifold shape (recursively).

Our objective is reachable only on the non-manifold 2D shapes [DFMMP03]. Instead, a non-manifold shape of higher-dimension is decomposed into several components, that belong to some computable superclasses of the manifolds. Thus, there exists an algorithm for recognizing the components of a given class. At this point, it is possible either to apply several *repairing* techniques [ACK13] on these components in order to obtain a decomposition with the desidered properties (but for a modified shape), or to consider the resulting decomposition of the original shape.

Mesh Repairing Applications. The first step of several *mesh repairing* applications [ACK13] is based on decomposing the non-manifold shapes into the collection of their subcomponents, connected along the non-manifold singularities. After this step, many techniques are applied in order to remove the non-manifold singularities, and to merge together several components in order to obtain a manifold shape from both the topological and the geometric point of view [Att10]. In this context, many techniques are defined for repairing the 2D and the 3D shapes.

One of the first approaches is in the context of the *Constructive Solid Geometry*, and is based on the *Boolean regularized* operators [TR80], that transform a 2D shape into a *regular* shape (containing only the 2D parts). In this context, a 2D shape is decomposed into its manifold subcomponents, not necessarily bidimensional, and their junctions are non-manifold (by construction). Intuitively, a 2D shape is the limit sequence of several manifolds, whose distance is given by the Haussdorf metric [DS92]. Specifically, this decomposition is represented by a hypergraph, such that every node corresponds to a component, and each arc corresponds to the singularities, shared by several components [FR92].

Similarly, Gueziec, et al. [GTLH98] propose a technique for repairing a non-manifold 2D shape by addressing only its topological and combinatorial aspects in the *cutting* and the *stitching* operations [Ago05]. The key idea for the cutting update consists of operating on proper copies of the singularities. A non-manifold shape is decomposed into several manifold components, and the majority of their border edges correspond to the original non-manifold edges. These components are merged pairwise along their border edges in order to form a new manifold complex (the stitching operation). No geometric aspect is considered, and the resulting complex may contain self-intersections, even if is manifold.

Rossignac and Cardoze introduce the *MatchMaker* algorithm [RC99], which solves the drawbacks of the Cutting & Stiching operations. They introduce small permutations in the Euclidean coordinates of the vertices in order to avoid the self-intersections and the unnecessary replications of the vertices. They show that it is possible to use a compact representation of the shape.

A similar approach is exploited in Attene et al. [AGFF09] for repairing a non-manifold 3D shape. Here, it is not possible to decompose a 3D shape into the collection of its manifold subcomponents without adding artificial cuts. In fact, the resulting components may contain some non-manifold singularities, that are fixed by removing a small portion for the neighborhood of the existing singularities. This operation is possible under several assumptions in both the combinatorial and the geometric sense.

Combinatorial Stratifications. The mesh repairing techniques are focused on removing the singularities and other defects from a shape. Instead, the singularities (if not removed) are the milestones for defining its structural model. In the current literature, there are many techniques, that decompose a non-manifold shape into a collection of its components, belonging to several dimension-independent and decidable superclasses of the manifolds.

Pesco et al. [PTL04] introduce the *Combinatorial Stratification* of the non-manifold 2D shapes, that are decomposed into several components, similar to the *strata* in the Whitney stratification for the analytic sets [Whi65]. The union of the strata covers the shape, and their pairwise intersection may be either empty, or common to both strata. The stratification is always decidable, and its strata are manifold, like in [GTLH98], but it is not unique. Following [DFH05], it is represented by a graph, such that its nodes correspond to the strata, and its arcs correspond to several copies of the non-manifold singularities. The strata are encoded by independent data structures, specific for manifold 2D shapes [Can12].

Lopes et al. [LNPT99] extend the *Combinatorial Stratification* to the 3D shapes. This stratification is not unique, and is represented by a graph-based data structure, similar to the 2D case. Every stratum may contain some singularities (as mentioned above), and is represented by the *Incidence Graph* [Ede87, Can12].

De Floriani et al. [DFMMP03] decompose a non-manifold shape of arbitrary dimension and not necessarily embedded in the Euclidean space as the collection of its *Initial Quasi-Manifold (IQM-)* components. These latter form the unique *IQM-decomposition* of a non-manifold shape. Here, the neighborhood of each vertex in an IQM-component of dimension k consists of one connected component, formed by several cells of dimension up to k. The IQM-components form a decidable and dimension-independent superclass of the manifolds. Specifically, they coincide with the 2D manifolds. Otherwise, they may contain non-manifold singularities in higher-dimension. Hui et al. [HVDF06] propose the *Double-Level Decomposition (DLD)* graph-based data structure for representing the IQM-decomposition of a non-manifold 3D shape, necessarily embedded in the Euclidean space $\mathbb{E}^3$, in the same spirit of the 2D and the 3D Combinatorial Stratifications. Here, the IQM-components are encoded by independent data structures [PBCF93].

1.2. Our Contribution

Our contribution may be considered as one of the first steps into the area of the automatic editing of the structural models, specific for a generic non-manifold shape Γ. In fact, many structural models are static (once created), and are not designed for being automatically updated, if the corresponding shape Γ is modified. A trivial solution consists of applying the update u of interest on Γ, and to obtain a new shape Γ_u. At this point, it is sufficient to recompute from scratch the decomposition $\mathcal{M}_{\Gamma_u}$ of the new shape Γ_u without considering the existing decomposition $\mathcal{M}_\Gamma$ of Γ. In the remainder of our paper, we indicate this approach as the *naive approach*. Here, we assume that it is possible to compute the decomposition of interest for a non-manifold shape. Thus, the naive approach is always computable, but it may be inefficient. In fact, an update u may modifiy only a *small* portion of Γ, and this means that only the corresponding components in $\mathcal{M}_\Gamma$ (modified by u) may be updated. Instead, the remaining parts of Γ (not modified by u) are unchanged. Thus, the corresponding components in $\mathcal{M}_\Gamma$ may be reused in $\mathcal{M}_{\Gamma_u}$. The resulting decomposition must be the same as the decomposition, computed by the *naive* approach. Hence, it is important to investigate when this simplification is feasible and under what constraints it is advantageous. This problem has not a unique solution, valid for all updates and all decompositions. A solution depends on the specific update and on how $\mathcal{M}_\Gamma$ is modified. In the current literature, there is a large number of the updates (also known as the *editing operators*), as shown in [CDF12].

In our paper, we limit our attention to the *Manifold-Connected (MC-)* decomposition of a non-manifold shape, initially proposed by Hui and De Floriani [HDF07] for simplicial 2D and 3D shapes. Specifically, we extend its definition to the cell shapes of arbitrary dimension. Its properties are still valid in this extension, and allow to exploit efficiently our approach. For the sake of simplicity, we show how our approach is exploited on the MC-decomposition of a 2D shape, if modified by the Euler operators in [LL01].

The remainder of our paper is organized as follows. Section 2 contains the background notions, that are used in this paper, while we extend the MC-decomposition in Section 3. In Section 4, we ad-

dress how the MC-decomposition is manipulated automatically by the Euler operators on the cell 2-complexes. In Section 5, we show how these operators are the basis for several geometry processing applications. In Section 6, we present our experimental results. Finally, we show the concluding remarks in Section 7.

2. Background Notions

In this section, we briefly propose some background notions. An interested reader finds more details in [Ago05, DFH05].

Euclidean cells and complexes. A *Euclidean k-cell* γ, embedded in the Euclidean space $\mathbb{E}^n$, is the homeomorphic image of $\mathcal{B}^k = \{x \in \mathbb{E}^k . \|x\| \leq 1\}$ with $0 \leq k \leq n$. Here, $k = dim(\gamma)$ is the *dimension* of the cell γ. For instance, a 0-cell is a vertex, a 1-cell is an edge, and a 2-cell is a polygon with an arbitrary number of vertices. A Euclidean cell d-complex Γ contains several cells of dimension up to d (and not necessarily of the same dimension) with disjoint interiors, such that the *boundary* $B(\gamma)$ of every k-cell γ (with $0 < k \leq d$) belongs to the union of the cells in Γ of dimension less than or equal to $(k-1)$. The maximum dimension d of its cells is the *dimension* of Γ. A d-cell in Γ is a *maximal* cell. We indicate the collection of all k-cells in Γ as Γ^k (for $0 \leq k \leq d$). A m-cell γ' in the d-complex Γ, such that $\gamma' \in B(\gamma)$, for a k-cell γ in Γ (for $0 \leq m < k \leq d$), is a *m-face* of γ. All $(k-1)$-faces in $B(\gamma)$ form the *immediate boundary* $B^i(\gamma)$ of γ. Similarly, a m-cell γ', such that $\gamma \in B(\gamma')$ (with $0 \leq k < m \leq d$), is a *m-coface* of γ. The *star* $St(\gamma)$ contains all m-cofaces of γ, for all $k < m$. All $(k+1)$-cofaces in $St(\gamma)$ forms the *immediate star* of the k-cell γ. A k-cell γ, such that $St(\gamma) = \emptyset$, is a *top* k-cell in Γ (with $0 \leq k \leq d$). We indicate the collection of all top k-cells in Γ as Γ_t^k. The top cells are not necessarily maximal cells. A d-complex Γ is *pure*, if and only if $\Gamma_t^d \equiv \Gamma^d \neq \emptyset$, and $\Gamma_t^k = \emptyset$, for all $0 \leq k < d$. Thus, all top d-cells in Γ are maximal cells. Instead, a not pure cell d-complex Γ contains several top cells, that are not maximal cells. Any two cells γ_1 and γ_2 (not both vertices) are *j-adjacent*, if they share a j-face, with $0 \leq j \leq min(dim(\gamma_1), dim(\gamma_2))$. Specifically, two k-cells γ_1 and γ_2 (with $k \neq 0$) are *adjacent*, if they share a $(k-1)$-face. Instead, two *adjacent* vertices are connected by an edge. The *carrier* $\Delta(\Gamma)$ of Γ is the subset of the Euclidean space, which is spanned by the union of its Euclidean cells. Following [Att10], a digital object O is described by the pair $\mathcal{O} = (\Gamma, \Delta(\Gamma))$ in order to separate the geometric aspects in $\Delta(\Gamma)$ from the connectivity aspects in Γ.

Combinatorial Characterization of Non-Manifolds. It is possible to characterize a k-cell γ as a *(combinatorial) manifold* cell in the d-complex Γ by analyzing its *link* [DFMMP03]. The *Link* $Lk(\gamma)$ of a k-cell γ consists of all cells, not in $St(\gamma)$, but belonging to the boundary of some cells in $St(\gamma)$. All edges (in red) in Figure 1(a) belong to the link $Lk(e_2)$ for the edge e_2. A d-complex is a *(combinatorial) manifold* d-complex, if $Lk(v)$, for each vertex v, is homeomorphic to the tessellation of either $\mathcal{S}^{d-1} = \{x \in \mathbb{E}^d . \|x\| = 1\}$, or to $\mathcal{D}^{d-1} = \{x \in \mathbb{E}^d . \|x\| < 1\}$, as shown in [Ago05]. A vertex v, at which this condition is violated, like the vertex v in Figure 1(b), is indicated as a *non-manifold singularity*, and the d-complex Γ is a *non-manifold* d-complex. Intuitively, a k-cell γ in the d-complex Γ is *manifold*, if $Lk(\gamma)$ is homeomorphic to the tessellation of either $\mathcal{S}^h$ or to D^{h-1}, for any $h \leq d-k-1$. This implies that a $(d-1)$-cell γ is a *manifold* cell in the d-complex Γ, if $St(\gamma)$ contains at most two maximal d-cells, like the edge e_2 in Figure 1(a).

The solution to this problem implies to check the existence of a tessellation for a given h-sphere [Nab96]. This operation is solved only for $h \leq 4$, and is reduced to the Halting Problem for $h \geq 6$. Its solution is still an open problem for $h = 5$. Hence, this operation is not always *computable* for any dimension, and a dimension-independent algorithm does not exist.

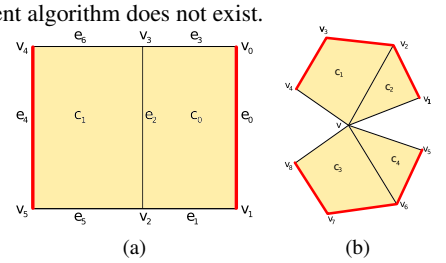

Figure 1: *Examples of the link (in red) for (a) the manifold edge e_2, and for (b) the non-manifold vertex v in two cell 2-complexes.*

3. Extending the Manifold-Connected (MC-) decomposition

In this section, we extend the *MC-decomposition*, initially proposed for the simplicial complexes [HDF07], to the cell d-complexes. A *simplicial d-complex* Σ is a cell d-complex Σ, such that its k-cells (called the *k-simplices*) are the linear combinations of $k+1$ affinely independent vertices, for $0 \leq k \leq d$. The intersection of two simplices is either empty, or a common simplex in Σ.

The Manifold-Connected (MC-) connectivity. A *k-path* $\gamma_1 \twoheadrightarrow_k \gamma_n$ is a sequence $(\gamma_i)_{i=1}^n$ of cells in a cell d-complex Γ (with $0 \leq k < d$), such that the cells γ_i and γ_{i+1} share a k-face τ_i. In the remainder of this paper, $\gamma_i \in (\gamma_1 \twoheadrightarrow_k \gamma_n)$ and $\tau_i \in (\gamma_1 \twoheadrightarrow_k \gamma_n)$ indicate that the cells γ_i and τ_i are traversed by (or belong to) the k-path $\gamma_1 \twoheadrightarrow_k \gamma_n$. A d-complex Γ, such that there is a k-path between any two cells is a *k-connected* complex. The cell 2-complex in Figure 1(b) is a 0-connected complex, since it exhibits at least two 0-paths (in red).

The $(k-1)$-paths, traversing the top k-cells in Γ_t^k (with $1 \leq k \leq d$), have a key role in the applications. Specifically, two top k-cells γ and γ' are *Manifold-Connected (MC-) adjacent*, if they are $(k-1)$-adjacent along a common $(k-1)$-face τ, and they are the unique top cells in $St(\tau)$. Two k-cells may be *multiply MC-adjacent* along more than one immediate face. For instance, the 2-cells c_0 and c_1 in Figure 2(a) are multiply MC-adjacent along the edges e_1, e_2, and e_3. The bidirectional arrows indicate two MC-adjacent cells.

Formally, we define the *MC-adjacency* relation $\xrightarrow{MC,k} \subseteq \Gamma_t^k \times \Gamma_t^k$ for two top k-cells γ and γ', such that $\gamma \xrightarrow{MC,k} \gamma'$ if:

$$\exists \tau \in \Gamma^{k-1} . \left(\tau \in B^i(\gamma) \cap B^i(\gamma')\right) \wedge \left(St^t(\tau) = \{\gamma, \gamma'\}\right)$$

Specifically, we say that a *MC-adjacency* occurs along the common face τ. By construction, the MC-adjacency relation is a reflexive and a symmetric relation, but it is not transitive. For instance, in Figure 2(b), the 2-cell q_1 is MC-adjacent to q_0, which is MC-adjacent to q_3, but the 2-cells q_1 and q_3 are not MC-adjacent. In any case, they belong to the 1-path $(q_i)_{i=0}^4$, such that all pairs of the consecutive 2-cells is MC-adjacent. A path with these properties is a *Manifold-Connected (MC-) path*, involving several top k-cells. Formally, a MC-path $\gamma_1 \twoheadrightarrow_{MC}^k \gamma_n$ is a $(k-1)$-path $(\gamma_i)_{i=1}^n$ in the d-complex Γ (for any $1 \leq k \leq d$), such that all pairs of the consecutive top k-cells γ_i and γ_{i+1} are MC-adjacent. A MC-path is the transitive closure of the MC-adjacency relation on the top k-cells in Γ.

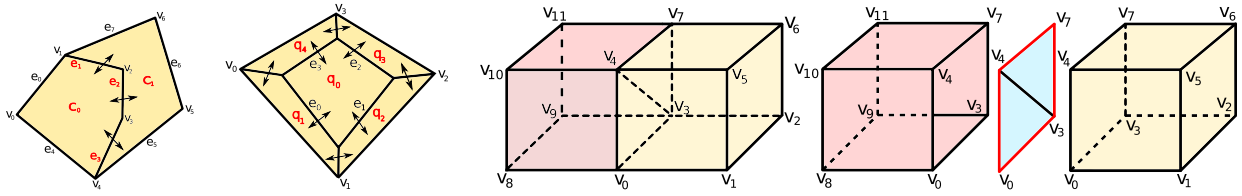

Figure 2: *(a) Example of the multiply MC-connected 2-cells c_0 and c_1. (b) A MC-complex of dimension 2 is defined by the MC-connectivity relation $\sim^2_{MC}$ on the top 2-cells. (c) The MC-decomposition of a 2-complex into three MC-components (in different colors) along some non-manifold edges (in red). The bidirectional arrows indicate a MC-adjacency between two top cells.*

Let γ be a top k-cell in a cell d-complex Γ (with $1 \leq k \leq d$). A *Manifold-Connnected (MC-) component* Γ^{MC}_{γ} of dimension k is the collection of the top k-cells in Γ, that belong to every MC-path, traversing the *representative cell* γ, i.e., all MC-paths of dimension k in Γ, outgoing from γ. Formally, a MC-component is described by the *MC-connectivity* relation $\sim^k_{MC} \subseteq \Gamma^k_t \times \Gamma^k_t$, which relates a top k-cell γ with any top k-cell γ', reachable through a MC-path, outgoing from γ. Here, $\gamma \sim^k_{MC} \gamma'$ if:

$$i)\ \exists \gamma' \in \Gamma^{MC}_{\gamma}.\gamma' \in \left(\gamma \twoheadrightarrow^k_{MC} \gamma'\right)$$

$$ii)\ \nexists \sigma \in \Gamma^k_t.(\sigma \neq \gamma) \wedge (\gamma' \in \Gamma^{MC}_{\sigma})$$

These conditions ensure that a MC-component (represented by the top k-cell γ) is uniquely determined, and is the maximal collection of all top k-cells, reachable from γ by a MC-path, that traverses γ. These cells are *MC-equivalent* to γ. It is clear that a top vertex v in Γ forms an independent MC-component $[v]$.

Lemma 1 Relation $\sim^k_{MC}$ is an equivalence relation, for $k \geq 1$.

Proof. By construction, a cell γ is MC-adjacent to itself, thus $\sim^k_{MC}$ is reflexive. Similarly, two top k-cells are mutually MC-adjacent, hence, $\sim^k_{MC}$ is also symmetric. Let $\gamma_1 \twoheadrightarrow^k_{MC} \gamma$ and $\gamma \twoheadrightarrow^k_{MC} \gamma_2$ be two MC-paths in Γ^{MC}_{γ}. For each $i = 1,2$, there exists an immediate face τ_i of γ, which is traversed by $\gamma \twoheadrightarrow^k_{MC} \gamma_i$. This implies that there exists a top k-cell γ'_i, such that $\gamma'_i \twoheadrightarrow^k_{MC} \gamma_i$, and $\gamma'_i \xrightarrow{MC,k} \gamma$ along τ_i. In other words, for each $i = 1,2$, a MC-adjacency occurs along τ_i, connecting γ and γ'_i, and there exists a MC-path from γ'_i to γ_i. Thus, there exists a MC-path $\gamma_1 \twoheadrightarrow^k_{MC} \gamma_2$. This shows that $\sim^k_{MC}$ is transitive, thus it is an equivalence relation. □

This result allows to define the MC-component Γ^{MC}_{γ} as the equivalence class $[\gamma] = \{\gamma' \in \Gamma^k_t.\gamma \sim^k_{MC} \gamma'\}$, for $k \geq 1$. Its representative top k-cell γ is arbitrarily chosen among the top k-cells, that are equivalent to γ with respect to the MC-connectivity relation. In the remainder, we will omit the dimension k in $\sim^k_{MC}$ (if no ambiguity arises), since k is implicitly determined by the representative cell of a MC-component. Intuitively, the Lemma 1 shows that the MC-paths are separated and composed together in order to form the new MC-paths of interest. This idea is the milestone of our approach, especially for validating the correctness of the implementations, and for simplifying the descriptions of the algorithms in Section 4.3.

Manifold-Connected complexes. A pure d-complex Γ, formed by only one MC-component $[\gamma]$, is a *MC-complex* of dimension d. By definition, the star of every $(d-1)$-cell τ contains one or two top d-cells, i.e., τ is manifold [DFMMP03]. Thus, a manifold d-complex is also a MC-complex of dimension d. Instead, a MC-complex is not necessarily manifold, since there may be some non-manifold p-cells with $0 \leq p \leq d-2$ [CDF13]. Thus, the class of the MC-complexes of dimension d is a superclass of the manifold d-complexes. These latter are not always computable [Nab96]. Instead, the class of the MC-complexes is always decidable for any dimension, since it is sufficient to compute the transitive closure of the MC-connectivity relation, as shown in [CDF11].

The Manifold-Connected (MC-) decomposition. The quotient space $\Gamma/\sim^d_{MC}$ of a pure d-complex Γ forms its *Manifold-Connected (MC-) decomposition* $\mathcal{MC}_{\Gamma}$. The equivalence classes correspond to the MC-components of Γ. For instance, Figure 2(c) shows the MC-decomposition of a pure 2-complex into three MC-components (in different colors).This approach is extended in two steps to a not pure d-complex Γ, containing a sparse distribution of the top k-cells, for every $0 \leq k \leq d$. First, the complex Γ is decomposed into the pure subcomplexes Γ^k_t, formed by all top k-cells in Γ. Then, for $k \geq 1$, the MC-decomposition $\mathcal{MC}_{\Gamma^k_t}$ is computed for every Γ^k_t. The *MC-decomposition* $\mathcal{MC}_{\Gamma}$ of the d-complex Γ consists of the top vertices in Γ, and of the union of all quotient spaces $\Gamma/\sim^k_{MC}$ for every $1 \leq k \leq d$. By construction, $\mathcal{MC}_{\Gamma}$ is unique, since every top k-cell γ belongs to only one Γ^k_t, thus to only one MC-component.

The Characterization of the Non-Manifold Singularities. For the sake of simplicitly, we say that the MC-component $[\gamma]$ belongs to the star $St(\tau)$ of any face τ, bounding γ. Thus, the number n^{MC}_{τ} of MC-components in $St(\tau)$ helps to recognize efficiently the non-manifold singularities [Can12]. Specifically, if $n^{MC}_{\tau} > 2$, then τ is non-manifold, as well as if $St(\tau)$ contains several MC-components of different dimension. Instead, the star of a manifold k-cell contains only two MC-components of dimension $k+1$. In any case, if $n^{MC}_{\tau} = 1$, then τ may be either manifold, or a *pinched* non-manifold singularity occurs at τ. In this latter case, face τ does not play a relevant role in the applications, and it can be discarded [BCMA*11].

Encoding the MC-decomposition. The MC-decomposition $\mathcal{MC}_{\Gamma}$ of a d-complex Γ is described by the two-level hypergraph $\mathcal{G}^{\Gamma}_{MC}$ [CDF13]. The lower level in $\mathcal{G}^{\Gamma}_{MC}$ contains a topological data structure $\mathcal{M}_{\Gamma}$, able to represent the non-manifold complex Γ. The *Generalized Indexed data structure with Adjacencies* [CDFW11] is one of the most compact representations for non-manifolds, as shown in [Can12]. The majority of the operations is performed in optimal time, and often in constant time, like the recognition of a MC-adjacency along a face τ, and the retrieval of n^{MC}_{τ}. Instead, the upper level of $\mathcal{G}^{\Gamma}_{MC}$ contains the MC-components in $\mathcal{MC}_{\Gamma}$. The layers of $\mathcal{G}^{\Gamma}_{MC}$ are independent, and decouple the local connectivity for the cells in Γ from the structural aspects in $\mathcal{MC}_{\Gamma}$.

4. The MC-decomposition and the Editing Operators

In this section, we propose the basic idea of our approach, if applied on a cell 2-complex, which is modified by the Euler operators, proposed in [LL01]. This is not a restriction, since these operators are the basis for all possible updates on a cell 2-complex.

4.1. Basic Idea of Our Approach

Following [DFPM97], an *update* $u = (u^-, u^+)$, applied on a complex Γ, removes the collection u^- of cells from Γ, and replaces them with the collection u^+ of new cells, that are added to $\Gamma/\{u^-\}$. The result is still a complex $\Gamma_u = \Gamma/\{u^-\} \cup \{u^+\}$ under some constraints. In this paper, we are interested in computing the MC-decomposition $\mathcal{MC}_{\Gamma_u}$ from $\mathcal{MC}_\Gamma$ without its complete recomputation (the *naive* approach). This is done in two steps. First, we identify what MC-components of $\mathcal{MC}_{\Gamma_u}$ are not *affected* (modified) by u, since they are reused directly in $\mathcal{MC}_{\Gamma_u}$. Then, the MC-components, affected by u, are modified, and saved in $\mathcal{MC}_{\Gamma_u}$.

Formally, we say that a MC-component $[\gamma]$ is *affected* by the update $u = (u^-, u^+)$, if at least one top cell γ' in $[\gamma]$ belongs to the *generalized neighborhood* $\sigma^h(u^-)$, for any order h [Att10]. Here, $\sigma^0(u^-)$ contains all top cells in the star of the vertices on the boundary of the cells in u^-. If $h \neq 0$, then $\sigma^h(u^-)$ contains all top cells in the star of the vertices, bounding the top cells in $\sigma^{h-1}(u^-)$. We consider the minimum order $\bar{h}$, such that $[\gamma]$ intersects $\sigma^{\bar{h}}(u^-)$ in order to have a unique generalized neighborhood to be considered. In any case, it is not mandatory that all top cells in $[\gamma]$ belong to $\sigma^{\bar{h}}(u^-)$. For instance, $\sigma^0(v_0) = \{q_1, q_4\}$ and $\sigma^1(v_0) = \{q_0, q_1, q_2, q_3, q_4\}$ for the vertex v_0 in the cell 2-complex in Figure 2(b). This latter is a MC-component, which intersects $\sigma^0(v_0)$, but q_0, q_2 and q_3 do not belong to $\sigma^0(v_0)$.

4.2. The Euler Operators and the MC-decomposition

Our approach could be applied on a complex of arbitrary dimension, and not necessarily embedded in the Euclidean space. For the sake of simplicity, we limit our attention to a generic cell 2-complex Γ with V vertices, E edges, and F polygons (i.e., the 2-cells). These latter are bounded by only one 0-connected path of edges. In particular, the *hole loops* (or the 1-*cycles*) play a key role in this context. Formally, the homology theory provides a precise definition of the 1-cycles [BCMA*11]. Intuitively, a hole loop is a closed 0-connected path of edges, such that the star of every edge is either empty, or contains several 2-cells, that are not bounded by all edges in the path. In other words, a hole loop encloses a void, and may become the immediate boundary of a 2-cell. Figure 3 shows a hole loop, formed by the top edges e, e_1, and e_3 and by the not top edge e_2. The complex Γ contains L hole loops and R connected components. Thus, the MC-decomposition $\mathcal{MC}_\Gamma$ may contain several MC-components of dimension up to 2.

We manipulate $\mathcal{MC}_\Gamma$ by using a simplified variant of the Euler operators in [LL01], satisfying the Euler equation $V - E + F = R - L$. In this case, there are five variables V, E, F, R, and L, and this means that five Euler operators (and their inverse operators) are needed. In addition, other two Euler operators are added for the sake of the commodity. Table 1 summarizes the behavior of these Euler operators in terms of the *Make (M)* and the *Kill (K)* operations, as well as the *Split (S)* and the *Join (J)* operations. Any Euler operators between the horizontal lines are mutually inverse.

Table 1: *The Euler operators in [LL01] are applied on a cell 2-complex with V vertices, E edges, F 2-cells, R regions, and L hole loops.*

	V	E	F	R	L
Make-Vertex-Region (MVR)	$+1$	$-$	$-$	$+1$	$-$
Kill-Vertex-Region (KVR)	-1	$-$	$-$	-1	$-$
Make-Edge-Loop (MEL)	$-$	$+1$	$-$	$-$	$+1$
Kill-Edge-Loop (KEL)	$-$	-1	$-$	$-$	-1
Make-Edge-Join-Region (MEJR)	$-$	$+1$	$-$	-1	$-$
Kill-Edge-Split-Region (KESR)	$-$	-1	$-$	$+1$	$-$
Make-Edge-Vertex (MEV)	$+1$	$+1$	$-$	$-$	$-$
Kill-Edge-Vertex (KEV)	-1	-1	$-$	$-$	$-$
Make-Face-Kill-Loop (MFKL)	$-$	$-$	$+1$	$-$	-1
Kill-Face-Make-Loop (KFML)	$-$	$-$	-1	$-$	$+1$
Split-Edge-Make-Vertex (SEMV)	$+1$	$+1$	$-$	$-$	$-$
Join-Edge-Kill-Vertex (JEKV)	-1	-1	$-$	$-$	$-$

4.3. The Actual Implementations of the Euler Operators

As shown in this section, the key operation for implementing the Euler operators of interest consists of merging and splitting efficiently the MC-components. The encoding of the MC-decomposition in Section 3 supports efficiently these operations. Only the unique identifier of the representative cell γ is encoded for each MC-component $[\gamma]$. The actual top cells in $[\gamma]$ belong to the transitive closure of $\sim_{MC}$, and are not needed when merging two MC-components, and splitting a MC-component. In fact, it is sufficient to update the representative cells. This operation is performed in constant time. The implementation of an update u is decoupled in two parts, involving the updates on $\mathcal{M}_\Gamma$ (that we do not consider), and the corresponding updates in $\mathcal{MC}_\Gamma$ (proposed in this section).

The MVR and the KVR operators. The MVR operator creates a new MC-component $[v]$, corresponding to a new top vertex v, thus to a new region in Γ. Instead, the KVR operator removes an existing top vertex v from Γ and the corresponding MC-component $[v]$ from $\mathcal{MC}_\Gamma$. The time complexity of both these operators is constant.

The MEL operator. This operator creates a new top edge e betwe en the vertices v_1 and v_n in the same region of Γ (see Figure 3). The vertices v_1 and v_n are not adjacent, and belong to a 0-path, formed by adjacent edges, such that their star is empty, or contains several 2-cells, that are not bounded by all edges in the 0-path. A new hole loop is created after adding e, but this is not relevant. Instead, it is mandatory to modify the MC-components in $St(v_1)$ and $St(v_n)$ after adding e. This latter belongs initially to the MC-component $[e]$, which is merged and manipulated only if $St(v_1)$ and $St(v_n)$ contain either only one or two top edges, as follows:

- if $St(v_1) = \{e'\}$, then it is possible to merge e with the MC-component in $St(v_1)$ by considering also $St(v_n)$:
  1. if $St(v_n) = \{e''\}$, then $[e]$ is merged with the MC-components in $St(v_1)$ and $St(v_n)$. If e' and e'' are MC-equivalent, then they belong to a MC-component $[e']$, which is replaced by $[e]$, containing e and the edges in $[e']$. Otherwise, they belong to $[e']$ and $[e'']$, respectively, that are replaced by $[e]$, containing e and the edges in $[e']$ and $[e'']$. The time complexity of these operations is linear in the number of the top edges in $[e']$.
  2. If $St(v_n) = \{e'', e^*\}$, then a not MC-adjacency occurs at v_n, and $St(v_n)$ contains a MC-component $[e'']$, which is split. In particular, a new MC-component $[e^*]$ is created, containing

all edges, that are MC-equivalent to e^*, and different from e''. The situation is the same as the previous case 1 (if we discard the edges in $[e^*]$). The time complexity of these operations is linear in the total number of the top edges in $[e']$ and $[e^*]$.

3. Otherwise, $[e]$ cannot be merged with other MC-components.

- If $St(v_1) = \{e', e^*\}$, then a not MC-adjacency also occurs at v_1, and $[e]$ is not merged with the MC-components in $St(v_1)$. By construction, e' and e^* belong to a MC-component, which is split into $[e']$ and $[e^*]$, containing, respectively, all top edges, that are MC-equivalent to e' and e^*. Then, $St(v_n)$ is checked as follows:
  - if $St(v_n) = \{e'', \bar{e}\}$, then a not MC-adjacency occurs at v_n, and the edges e'' and $\bar{e}$ belong to a MC-component, which is removed and split into two MC-components $[e'']$ and $[\bar{e}]$. These latter contain, respectively, all top edges, that are MC-equivalent to e'' and $\bar{e}$. Then, $[e']$ and $[e'']$, as well as $[e^*]$ and $[\bar{e}]$, are merged pairwise (respectively), if their representative top edges are MC-equivalent. The time complexity of this operation is linear in the number of the edges in $[e']$ and $[e^*]$.
  - Otherwise, there is nothing to be done.
- If the roles of v_1 and v_n in the previous cases are swapped, then the resulting situation is the same.

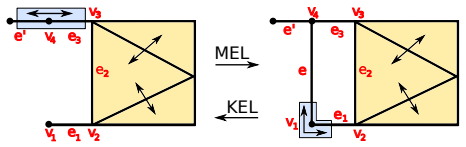

Figure 3: *The MEL and the KEL operators modify a top edge e in the hole loop, formed by the edges e, e_1, e_2, and e_3 in a 2-complex.*

The KEL operator. This operator removes an existing top edge $e = (v_1, v_n)$ from a hole loop $\{e_1, \ldots, e_{n-1}, e\}$ in the 2-complex Γ. The star of every edge in the hole loop may be top, or may contain several 2-cells, that are not bounded by all edges in the hole loop. After removing e, the remaining edges form a not closed 0-path $v_1 \nrightarrow_0 v_n$. By construction, e belongs to an existing MC-component $[e]$, from which it is removed. In order to modify correctly the MC-components of dimension 1 in $St(v_1)$ and $St(v_n)$, it is important to check whether a not MC-adjacency continues to occur at v_i, even if e is removed (for $i = 1$ and $i = n$), i.e., if $St(v_i)$ contains at least one MC-component of dimension 2, and at least four top edges (including e). Otherwise $St(v_i)$ contains e, and up to two top edges. We proceed as follows (see Figure 3):

- if a not MC-adjacency occurs both at v_1 and v_n, then $[e]$ is removed. The time complexity of this operation is constant.
- Otherwise, we check $St(v_1)$ and $St(v_n)$ as follows:
  1. if $St(v_1) = \{e, e'\}$, then $St(v_1)$ contains only the MC-component $[e]$ (formed by at least e and e'), becoming $[e']$. If possible, $[e']$ is merged with an existing MC-component, formed by a MC-path $v_1 \nrightarrow^1_{MC} v_n$, which does not traverse e. The time complexity of these operations is linear in the number of the top edges, that are MC-equivalent to e'.
  2. If $St(v_1) = \{e', e'', e\}$, then a MC-adjacency occurs at v_1 after removing e, and $[e']$ and $[e'']$ are merged together. The resulting MC-component is indicated as $[e']$. At this point:
     - if $St(v_n) = \{\bar{e}', \bar{e}'', e\}$, then it possible to merge together the existing MC-components $[\bar{e}']$ and $[\bar{e}'']$ in $St(v_n)$ in order to obtain a new MC-component $[\bar{e}']$ (as in the previous case). If the edges in $[e']$ are MC-equivalent to the edges in $[\bar{e}'']$, then they are merged into a new MC-component $[\bar{e}'']$, otherwise $[e']$ and $[\bar{e}'']$ remain independent.
     - If $St(v_n) = \{\bar{e}', e\}$, then a new MC-component $[\bar{e}']$ is merged with $[e']$ (like in the case 1), if possible.
     - If a not MC-adjacency occurs at v_n, even if e is removed, then $[e']$ is merged with an existing MC-component in $St(v_n)$, if possible.

     The time complexity of these operations is linear in the number of the top edges in $[e']$.
  3. If the roles of v_1 and v_n in the previous cases are swapped, then the resulting situation is the same.

These operators update many MC-components, intersecting the generalized neighborhood $\sigma^0(e)$, but the top edges of interest do not belong necessarily to $\sigma^0(e)$.

The MEJR and the KESR operators. The MEJR operator creates a new top edge e between the existing vertices v_1 and v_n, belonging, respectively, to two disconnected regions of Γ, that are merged into one connected region. Instead, the KESR operator removes an existing top edge $e = (v_1, v_n)$, connecting two regions of Γ. These latter are disconnected, if e is removed. These operators (see Figure 4) are similar to the MEL and the KEL operators, but without manipulating the hole loops.

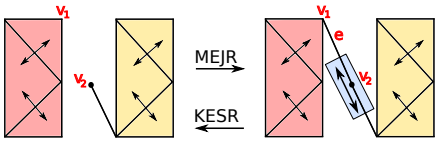

Figure 4: *The MEJR and the KESR operators modify a top edge $e = (v_1, v_2)$, connecting two regions in a 2-complex.*

The MEV operator. This operator creates a new top edge $e = (v, v')$ between an existing vertex v and a new vertex v' in Γ. Edge e represents the MC-component $[e]$, which may be merged with a MC-component of dimension 1. It is necessary to check the existence of a MC-adjacency only along v, since v' is new. In particular:

- if v is a top vertex in Γ, then it represents the MC-component $[v]$. Thus, $[e]$ is added as a new MC-component, and $[v]$ is removed from $\mathcal{MC}_\Gamma$. The time complexity of these operations is constant.
- Otherwise, the existing MC-components in $St(v)$ in addition to $[e]$ are analyzed as follows (see Figure 5):
  1. if $St(v)$ contains either at least one MC-component of dimension 2, or at least two MC-components, not necessarily of the same dimension, then it is not possible to merge $[e]$ with other MC-components in $\mathcal{MC}_\Gamma$. Thus, $[e]$ is added to $\mathcal{MC}_\Gamma$, and also the time complexity of these operations is constant.
  2. Otherwise, $St(v)$ contains only one MC-component $[e']$ of dimension 1, containing either one top edge e', or two top edges e' and e'' in $St(v)$, by construction (see Figure 5). Hence:
     - if $St(v) = \{e'\}$, then a new MC-component is created, containing e and the top edges in $[e']$. For the sake of simplicity, the new MC-component is represented as $[e]$, and the time complexity of these operations is constant.

– If $St(v) = \{e', e''\}$, then it is not possible to merge $[e]$ with $[e']$, since a MC-adjacency cannot occur at v. The new MC-component $[e]$ is added to $\mathcal{MC}_\Gamma$, and it is necessary to check if e' and e'' belongs to a closed MC-path, connecting v with itself without traversing e. In other words, if $[e']$ is a hole loop, formed only by several top edges, then $[e']$ remains unchanged in $\mathcal{MC}_\Gamma$. Otherwise, $[e']$ is removed from $\mathcal{MC}_\Gamma$, and is split into two MC-components, containing all top edges, that are, respectively, MC-equivalent to e' and e'' (see Figure 5). These new MC-components are indicated as $[e']$ and $[e'']$. The time complexity of these operations is linear in the number of the top edges in $[e']$.

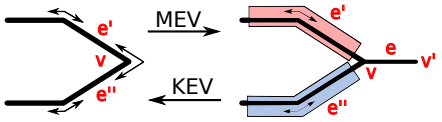

Figure 5: *The MEV and the KEV operators modify a top edge e in the star of a vertex v in a cell 2-complex.*

The KEV operator. This operator removes an existing top edge $e = (v, v')$, as well as the vertex v', such that $St(v') = \{e\}$. By construction, e belongs to an existing MC-component $[e]$. This MC-component is modified as follows (see Figure 5):

1. if $St(v) = \{e\}$, then $[e]$ is removed from $\mathcal{MC}_\Gamma$, since it contains no top edge. After removing e, the vertex v becomes a top vertex, and a new MC-component $[v]$ is added to $\mathcal{MC}_\Gamma$. Hence, the time complexity of these operations is constant.
2. If $St(v)$ contains either at least one MC-component of dimension 2, or at least three MC-components, not necessarily of the same dimension, in addition to $[e]$, then $[e]$ is removed from $\mathcal{MC}_\Gamma$. The time complexity of these operations is constant.
3. If $St(v)$ contains two MC-components $[e']$ and $[e'']$ of dimension 1 in addition to $[e]$, then a MC-adjacency occurs at v, after removing e from Γ and $[e]$ from $\mathcal{MC}_\Gamma$. For the sake of simplicity, we assume that e' and e'' are in $St(v)$. Hence, the MC-components $[e']$ and $[e'']$ are removed from $\mathcal{MC}_\Gamma$, and a new MC-component $[e']$ (which may be also indicated as $[e'']$) is added to $\mathcal{MC}_\Gamma$. The new MC-component $[e']$ contains the union of the top edges in the previous MC-components $[e']$ and $[e'']$. The time complexity of these operations is constant.
4. If $St(v)$ contains only two top edges e and e', belonging to the same MC-component $[e]$ (by construction), then it is sufficient to remove $[e]$ from $\mathcal{MC}_\Gamma$, and to add a new MC-component $[e']$, containing all top edges in the previous MC-component $[e]$, except e. The time complexity of these operations is constant.

The MFKL operator. This operator creates a new 2-cell γ, which fills the void, enclosed by an existing hole loop $\{e_i\}_{i=1}^n$ in Γ. After adding γ, the hole loop is removed, and the star of each edge e_i contains at least γ. Thus, all MC-components, formed by edges in the hole loop, are removed. At this point, it is necessary to check if γ may be merged with the existing MC-components along any edge in the hole loop. We store their representative cells in a set s, initially empty, and we proceed on each e_i as follows (see Figure 6):

1. if edge e_i is top, then it becomes a border edge of γ.
2. If $St(e_i) = \{\gamma'\}$, then it is possible to merge γ with the existing MC-component $[\gamma']$ of dimension 2. The 2-cell γ' is stored in s.
3. If $St(e_i)$ contains at least three top 2-cells in addition to γ, then a not MC-adjacency occurs at e_i.
4. If $St(e_i) = \{\gamma', \gamma''\}$ (except γ), then γ' and γ'' belong to an existing MC-component, which may remain unchanged, or be split into two MC-components. Let $b_{\gamma'}$ be a boolean flag, which is true, if and only if there exists a MC-path $\gamma' \twoheadrightarrow^2_{MC} \gamma$, not traversing γ'', but passing through an edge $e_{\gamma'}$ in the initial hole loop. This MC-path identifies a new MC-component $[\gamma']$. Similarly, let $b_{\gamma''}$ be a boolean flag, which is true, if and only if there exists a MC-path $\gamma'' \twoheadrightarrow^2_{MC} \gamma$, not traversing γ', but passing through an edge $e_{\gamma''}$ in the initial hole loop. This MC-path identifies a new MC-component $[\gamma'']$. If $e_{\gamma'} = e_{\gamma''}$, then the existing MC-component in $St(e_i)$ is split into two MC-components $[\gamma']$ and $[\gamma'']$. Otherwise, it is necessary to check $b_{\gamma'}$ and $b_{\gamma''}$ as follows:
   - if both $b_{\gamma'}$ and $b_{\gamma''}$ are true, then the MC-component $[\gamma']$ in $St(e_i)$ is merged with γ, and γ' is added to s.
   - If both $b_{\gamma'}$ and $b_{\gamma''}$ are not true, then the MC-component of interest in $St(e_i)$ is removed, and replaced by $[\gamma']$ and $[\gamma'']$.
   - If only $b_{\gamma'}$ is true, then the MC-component of interest in $St(e_i)$ is removed, and split into $[\gamma']$, which is merged with γ (γ' is saved in s), and $[\gamma'']$, which is directly created in $\mathcal{MC}_\Gamma$.
   - If only $b_{\gamma''}$ is true, then the MC-component of interest in $St(e_i)$ is removed, and split into $[\gamma'']$, which is merged with γ (γ'' is saved in s), and $[\gamma']$, which is directly created in $\mathcal{MC}_\Gamma$.

A new MC-component $[\gamma]$ is formed by 2-cells in s. The time complexity of these operations is linear in the number of the 2-cells in the MC-components, that are incident at the edges in the hole loop.

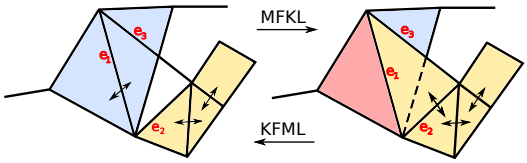

Figure 6: *The MFKL and the KFML operators modify a 2-cell, which is bounded by the hole loop of edges e_1, e_2, and e_3.*

The KFML operator. This operator removes an existing 2-cell γ without destroying its immediate edges $\{e_i\}_{i=1}^n$, that form a new hole loop in Γ. It is important to check if new MC-components of dimension 1, involving all edges in $B^i(\gamma)$, that become top edges, after removing γ, must be added to $\mathcal{MC}_\Gamma$. We assume to store these edges into a list q, initially empty. It may be necessary to fix and to update correctly those MC-components, that are connected initially through γ. We assume to store the representative 2-cells of these MC-components into a set s, initially empty. The MC-components in $St(e_i)$, with $i = 1, \ldots, n$, are updated as follows (see Figure 6):

1. if $St(e_i) = \{\gamma\}$, then e_i becomes a top edge after removing γ, and is saved in the list q.
2. If $St(e_i)$ contains more than three top 2-cells, then a not MC-adjacency continues to occur at e_i even if γ is removed.
3. If $St(e_i) = \{\gamma, \gamma'\}$, then a MC-adjacency occurs at e_i. By construction, γ and γ' belongs to an existing MC-component, which may be split when removing γ. The resulting MC-component is indicated as $[\gamma']$, and γ' is saved in s.
4. If $St(e_i) = \{\gamma, \gamma', \gamma''\}$, then a new MC-adjacency occurs at e_i between γ' and γ'', when γ is removed, corresponding to a new

MC-component $[\gamma']$. This latter contains all 2-cells, that are MC-equivalent to γ' (saved in s) and γ''.

The time complexity of this traversal is linear in n. At this point, the set s contains the representative cells of those MC-components, traversing initially γ, that may be modified (if necessary). At this point, every 2-cell γ_i in s is visited as follows:

- if there does not exist $\gamma_i \twoheadrightarrow^2_{MC} \bar{\gamma}$ for any $\bar{\gamma}$ in s, then γ_i is removed from s, and a new MC-component $[\gamma_i]$ is created.
- Otherwise, there exists a MC-path, connecting γ_i with several 2-cells in s. These latter and γ_i form a new MC-component $[\gamma_i]$, and are removed from s.

The remaining 2-cells in s form a new MC-component. The time complexity of these operations is linear in the total number of the 2-cells in the MC-components, that are incident at the edges in $B^i(\gamma)$. Finally, we identify all MC-components of dimension 1, formed by all top edges in q. These new MC-components are added to $\mathcal{MC}_\Gamma$. The time complexity of these operations is linear in $\|q\|$.

The SEMV and the JEKV operators. The SEMV operator creates a new vertex v, and splits a top edge $e = (v_1, v_2)$ into the top edges $e_1 = (v_1, v)$ and $e_2 = (v_2, v)$. The edge e belongs to the MC-component $[e]$, e_1 and e_2 correspond, respectively, to $[e_1]$ and $[e_2]$, and a MC-adjacency occurs at v. Thus, e_1 and e_2 belong to $[e]$ (see the Lemma 1), becoming $[e_1]$. Instead, the JEKV operator is the reverse operator, and joins $e_1 = (v_1, v)$ and $e_2 = (v, v_2)$ into the edge $e = (v_1, v_2)$. The time complexity of these operators is constant.

5. The MC-decomposition and Other Updates

In this section, we show how applying several high-level updates on the MC-decomposition $\mathcal{MC}_\Gamma$ of a 2-complex Γ.

Template-based Updates. These updates are based on a specific template pattern, applied on each 2-cell in the same spirit of the interactive retopology [MTP*15], the polyhedral patterns [JTV*15], as well as the Stellar and the Handle operators [CDF12]. Broadly speaking, these operators replace a top 2-cell with any of its possible tessellations. In other words, they replace an existing MC-path with another MC-path, such that the new top cells cover the original cells. Following Lemma 1, this operation is closed with respect to $\sim_{MC}$ in the same spirit of the SEMV and the JEKV operators. These updates are implemented by combining the KFML, the MEV, and the MEL operators, as well as their inverse operators. Some examples are shown in Figures 2. The template-based operators are also defined on the 3-complexes [OS15], and the stellar operators are dimension-independent.

Merging two MC-adjacent 2-cells. This update merges two 2-cells γ_1 and γ_2, that are MC-adjacent along the edges $\{e_i\}_{i=1}^n$, into a new 2-cell γ in Γ. This operation removes γ_1 and γ_2 through the KFML operator, which creates two loops. The edge e_1 is removed by the KFML operator (destroying two loops), and the remaining edges $\{e_i\}_{i=2}^n$ are deleted through the KEV operator, creating a new (larger) hole loop. This latter is the immediate boundary of the new 2-cell γ, added by the MFKL operator. The reverse operator splits an existing 2-cell into two 2-cells, and is a template-based operator.

These editing operators modify only the connectivity of the top cells in a MC-component, and $\sim_{MC}$ is closed with respect to these updates. Thus, $\mathcal{MC}_\Gamma$ remains unchanged. Other updates, e.g., the cell and edge collapses, may modify $\mathcal{MC}_\Gamma$.

Cell collapse. This operator is common in the homology computations [BCMA*11], and collapses a top cell along an immediate border face. It is equivalent to the KEV operator, if applied on a top edge. Instead, it removes a 2-cell through the KFML operator, and a border edge through the MEL operator.

The Edge Collapse. This update collapses an existing edge $e = (v_1, v_2)$ into a vertex v in the 2-complex Γ. Broadly speaking, all 2-cells in $St(e)$ are removed, and all cells, belonging only to $St(v_1)$ or $St(v_2)$ are made incident at v. The collapse of a manifold edge is expressed in terms of the Stellar operators [Ago05]. In the case, if the 2-cells in $St(e)$ are completely contained in a MC-component, and no edge in their immediate boundaries is not manifold, then $\sim_{MC}$ is closed with respect to the edge collapse (see Figure 7(a)), since the collapse becomes a specific template-based operator (in this case). Also the inverse operator, called the *Vertex Split (VS)* operator satisfies the same conventions. Instead, $\sim_{MC}$ is not closed with respect to the collapse of either a top (see Figure 7(b)) or a non-manifold edge (see Figure 7(c)). In fact, several MC-components may be merged into only one MC-component. Similarly, this operator may either reduce the dimension of the top cells in a MC-component, or split a MC-component into new MC-components of different dimension. In other words, the edge collapse modifies not only the connectivity of the cells, but also the topology of Γ. Thus, it is mandatory to reconstruct a large number of the MC-components, that intersect $\sigma^h(e)$, for any $h >> 0$, corresponding even to Γ. Collapsing an edge without satisfying specific constraints is not efficient. Currently, updating the MC-decomposition with an edge collapse is an open problem, and must be investigated in the future.

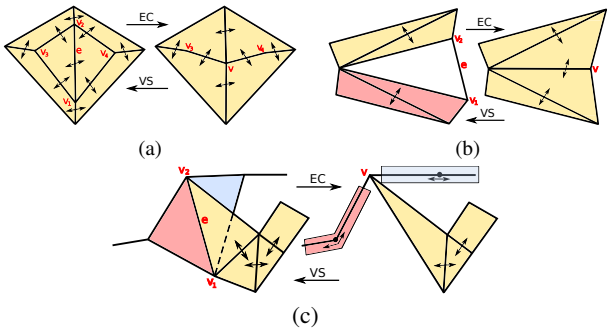

Figure 7: *(a) The collapse of a manifold edge, such that its star is stored in a MC-component, and none of their edges is non-manifold, maintains $\sim_{MC}$. This is not true, if collapsing (b) a top, and (c) a non-manifold edge.*

6. Experimental Results

In this section, we show our experimental results, that are based on [CDF13]. In any case, our implementation is designed on a *mangrove*, which is the graph-based representation of a topological data structure [CDF14]. The mangroves are the basis for the *Mangrove Topological Data Structure (Mangrove TDS) Library* [MAN], focused on the fast design of the topological data structures. Hence, our implementations are written only once, and reused with several topological data structures, loaded *dynamically*. To the best of our experience, this is the unique method in the current literature for reaching this objective without the wrapping techniques. Our implementations will be publicly available.

In order to validate our approach, we evaluate the running times,

Model	s^t	n^{MC}	m	s^t_m	$\mathcal{T}^N_m$	$\mathcal{T}^B_m$	$\mathcal{T}^I_m$	m	s^t_m	$\mathcal{T}^B_m$	$\mathcal{T}^I_m$	m	s^t_m	$\mathcal{T}^B_m$	$\mathcal{T}^I_m$
Cubes (quad.)	40.4K	4	1K	44.4K	19.1K	19	0.55	100K	440.4K	189	51	1.5M	6M	4.39K	823
			5K	60.4K	115.7K	28	2.8	200K	840K	371	107	2M	8M	5.4K	1.08K
			10K	80.4K	281.4K	35	5	500K	2M	1K	360	2.5M	10M	7.2K	1.39K
			40K	220.4K	–	85	21.4	1M	4M	1.87K	520.6	3M	12M	8.6K	1.67K
Frame (quad.)	40.7K	132	1K	44.7K	19.9K	21	0.58	100K	440.7K	196	58	1.5M	6.1M	4.8K	830
			5K	60.7K	115.9K	29	3	200K	840.7K	395	111	2M	8.1M	5.8K	1.1K
			10K	80.7K	294K	36	5.5	500K	2.1M	1.1K	372	2.5M	10.1M	7.3K	1.49K
			40K	200.7K	–	86	22	1M	4.1M	1.9K	538	3M	12.1M	8.7K	1.68K
Twist (simpl.)	82.4K	4	5K	92.4K	183K	41	2.8	200K	482.4K	842	107	3M	6M	10.8K	1.67K
			10K	102.4K	392K	48	5.3	500K	1M	2.4K	360	4M	8M	11.8K	2.1K
			40K	162.4K	–	163	21	1M	2M	3.7K	521	5M	10M	16.8K	2.8K
			100K	286.4K	–	394	51.2	2M	4M	8.3K	1.08K	6M	12M	23.4K	3.3K
Mecha (simpl.)	86.7K	45	5K	96.7K	195K	42	2.85	200K	486.7K	843.1	107.3	3M	6.1M	10.9K	1.68K
			10K	106.7K	413K	49	5.4	500K	1.1M	2.42K	362.9	4M	8.1M	11.9K	2.16K
			40K	166.7K	–	166	23	1M	2.1M	3.71K	521.7	5M	10.1M	16.9K	2.9K
			100K	286.7K	–	395	52.1	2M	4.1M	8.4K	1.09K	6M	12.1M	23.5K	3.4K

Table 2: *The experimental results show the validity of our interactive approach (see the running time $\mathcal{T}^I_m$), after applying m random updates on the MC-decompositions (with n^{MC} MC-components) for several 2-complexes with s^t_m cells (initially $s^t = s^t_0$), with respect to the running times $\mathcal{T}^N_m$ and $\mathcal{T}^B_m$ of the* naive *and the* batch *approaches, respectively. First two shapes are discretized by non-manifold quad meshes and updated by the template-based operators, while the remaining shapes are described by simplicial 2-complexes and updated by the stellar split operators. All running times are expressed in milliseconds.*

necessary for computing the MC-decomposition $\mathcal{MC}_\Gamma$ after applying m random updates on a 2-complex Γ. The time complexity for computing $\mathcal{MC}_\Gamma$ is linear in s^t [Can12]. First, we analyze the running time $\mathcal{T}^N_m$, needed for computing $\mathcal{MC}_\Gamma$ by using the *naive* approach. After applying k random updates (for any $0 \leq k \leq m-1$), the resulting complex Γ_k contains $s^{t,k}$ top cells. Thus, $\mathcal{T}^N_m$ is linear in $\sum_{k=0,\ldots,m-1} s^{t,k}$. We also evaluate the running time $\mathcal{T}^B_m$ of the *batch* approach, which is used for computing $\mathcal{MC}_\Gamma$ only after applying m updates on Γ. Finally, we evaluate the total running time $\mathcal{T}^I_m$, needed for applying interatively m random updates on Γ. We apply m random stellar (split) updates on the top simplices, and m random template-based updates on the top cells in a cell 2-complex. This latter contains initially s^t top cells. This is not a restriction, since the implementation of these updates requires almost the Euler operators in Section 4.3. Table 2 shows our experimental results, obtained on a workstation with Intel i5 processor and 4Gb RAM. We compare the running times $\mathcal{T}^N_m$, $\mathcal{T}^I_m$, and $\mathcal{T}^B_m$ (in milliseconds) by varying m. The running times, requiring more than 10 minutes of computations (6×10^5 ms), are discarded. It is clear that the *naive* approach may be unfeasible, even if after a small number m of updates (about 40K). Already in this case, $\mathcal{T}^N_m$ exceeds 10 minutes of computation. Our interactive approach exhibits fast running times $\mathcal{T}^I_m$ (even at interactive rate), and is also more competitive than the *batch* approach. Thus, it provides a more efficient strategy for constructing (interactively) the MC-decomposition $\mathcal{MC}_\Gamma$.

7. Conclusions and Future Work

This paper provides one step towards the automatic editing of the structural model for the non-manifold shapes in the same spirit of [IS15]. We have addressed the problem of manipulating automatically the MC-decomposition of a complex [HDF07] through the Euler operators without being recomputed from scratch after an update. This approach is feasible, since only a local portion of a complex is modified by an update. It is an important advantage with respect to recomputing completely the MC-decomposition. In this paper, we have focused our attention only on the 2-complexes, but our approach can be generalized in any dimension. Our implementations are based on the two-level graph-based representation of the MC-decomposition in [CDF13], integrated with a *mangrove*. This latter is the graph-based representation for a topological data structure [CDF14], and is the basis of the *Mangrove TDS Library* [MAN], focused on the fast design of the topological data structures. Our implementation is designed only once, and reused with several topological data structures.

Our experimental results show that the manipulation of the MC-decomposition, currently considered as a *batch* task, can be solved at *interactive rate*, like the applications in [JGGN15,MTP*15]. Our approach also allows to to improve the computational cost of several applications, e.g., the semantic reasoning [HDF07] and the homology computations [BCMA*11], by reducing the resolution, and maintaining the structure for the complex of interest. Another application is in the context of the 3D printing, like in [WZK16]. Other possible applications try to optimize the quality of the elements in the MC-components, satisfying a specific criterion in the same spirit of [Si15] (just to mention one). These techniques are based on the Euler operators, discussed in this paper.

Finally, our contribution is an intermediate step towards the *multiresolution structural model* of a non-manifold shape, which takes into account the dependency relation among the updates, applied on the MC-decomposition, like in [DFPM97]. Specifically, it is important to consider what MC-components and top cells are created/removed by an update, as well as the connectivity of the MC-components along the non-manifold singularities. This model is challenging, but is simplified in the same spirit of [CADM11]. In other words, it is not necessary to encode explicitly the connectivity of the top-cells in a MC-component, since they are MC-equivalent. It is also possible to encode only the dependency relation among the MC-components, and their connectivity through the non-manifold singularities. Instead, the explicit connectivity of the top cells in the MC-components is generated on-the-fly upon request by the

automatic mesh generation, the retopology, and the quadrangulation techniques (just to mention some). This allows to extract the structural representations at different resolutions without considering the combinatorial and the geometric aspects of a complex.

References

[ACK13] ATTENE M., CAMPEN M., KOBBELT L.: Polygon Mesh Repairing: Application Perspective. *ACM Computing Surveys 45*, 2 (2013). 1

[AGFF09] ATTENE M., GIORGI D., FERRI M., FALCIDIENO B.: On Converting Sets of Tetrahedra to Combinatorial and PL Manifolds. *Computer-Aided Geometric Design 26*, 8 (2009), 850–864. 2

[Ago05] AGOSTON M.: *Computer Graphics and Geometric Modeling*. Springer, 2005. 2, 3, 8

[Att10] ATTENE M.: A Lightweight Approach to Repairing Digitized Polygon Meshes. *The Visual Computer 26*,11 (2010), 1393–1406. 1, 3, 5

[BCMA*11] BOLTCHEVA D., CANINO D., MERINO ACEITUNO S., LEON J.-C., DE FLORIANI L., HETROY F.: An Iterative Algorithm for Homology Computation on Simplicial Shapes. *Computer-Aided Design 43*, 11 (2011), 1457–1467. 1, 4, 5, 8, 9

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LËVY B.: *Polygon Mesh Processing*. A. K. Peters, 2010. 1

[CADM11] CASTELLI ALEARDI L., DEVILLERS O., MEBARKI A.: Catalog-based Representations of 2D Triangulations. *International Journal of Computational Geometry and Applications 21*, 4 (2011), 393–402. 9

[Can12] CANINO D.: *Tools for Modeling and Analysis of Non-manifold Shapes*. PhD thesis, Department of Computer Science, University of Genova, Italy, 2012. 1, 2, 4, 9

[CDF11] CANINO D., DE FLORIANI L.: A Decomposition-based Approach to Modeling and Understanding Arbitrary shapes. In *Proceedings of the Eurographics Italian Chapter Conference* (2011), pp. 53–60. 4

[CDF12] COMIC L., DE FLORIANI L.: Modeling and Manipulating Cell Complexes in Two, Three and Higher Dimensions. In *Digital Geometry Algorithms: Theoretical Foundations and Applications to Computational Imaging*. Springer, 2012. 2, 8

[CDF13] CANINO D., DE FLORIANI L.: A Compact Representation for Topological Decompositions of Non-manifold Shapes. In *Proceedings of the International Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications* (2013), pp. 100–107. 4, 8, 9

[CDF14] CANINO D., DE FLORIANI L.: Representing Simplicial Complexes with Mangroves. In *Proceedings of the International Meshing Roundtable* (2014), Springer, pp. 465–483. 8, 9

[CDFW11] CANINO D., DE FLORIANI L., WEISS K.: IA*: An Adjacency-based Representation for Non-Manifold Simplicial Shapes in Arbitrary Dimensions. *Computer & Graphics 35*, 3 (2011), 747–753. 4

[DFH05] DE FLORIANI L., HUI A.: Data Structures for Simplicial Complexes: Analysis and Comparisons. In *Proceedings of the Symposium of Geometry Processing* (2005). 1, 2, 3

[DFMMP03] DE FLORIANI L., MESMOUDI M., MORANDO F., PUPPO E.: Non-manifold Decompositions in Arbitrary Dimensions. *Geometric Models*, 65 (2003), 2–22. 1, 2, 3, 4

[DFPM97] DE FLORIANI L., PUPPO E., MAGILLO P.: A Formal Approach to Multiresolution Modeling. In *Geometric Modeling: Theory and Practice*. 1997, pp. 302–323. 5, 9

[DS92] DESAULNIERS H., STEWART N.: An Extension of the Manifold Boundary Representations to the R-sets. *ACM Transactions on Graphics 11*, 1 (1992), 40–60. 2

[Ede87] EDELSBRUNNER H.: *Algorithms in Combinatorial Geometry*. Springer, 1987. 2

[FR92] FALCIDIENO B., RATTO O.: Two-Manifold Cell Decomposition of the R-sets. *Computer Graphics Forum 11*, 3 (1992), 391–404. 2

[GTLH98] GUEZIEC A., TAUBIN G., LAZARUS F., HORN W.: Converting Sets of Polygons to Manifold Surfaces by Cutting and Stitching. In *Proceedings of the IEEE Conference on Visualization* (1998), pp. 383–390. 2

[HDF07] HUI A., DE FLORIANI L.: Two-level Topological Decomposition for Non-Manifold Simplicial Shapes. In *Proceedings of the Conference on Solid and Physical Modeling* (2007), pp. 355–360. 2, 3, 9

[HVDF06] HUI A., VACZLAVIK L., DE FLORIANI L.: Decomposition-based Representation for 3d Simplicial Complexes. In *Proceedings of the Symposium on Geometry Processing* (2006), pp. 101–110. 2

[IS15] IURICICH F., SIMARI P.: Hierarchical Mesh Segmentation Editing through Rotation Operations. In *Proceedings of the Conference on Image Analysis and Processing* (2015), pp. 398–409. 9

[JGGN15] JIN M., GOPSTEIN D., GINGOLD Y., NEALEN A.: AniMesh: Interleaved Animation, Modeling, and Editing. *ACM Transactions on Graphics 34*, 6 (2015). 9

[JTV*15] JIANG C., TANG C., VAXMAN A., WONKA P., POTTMANN H.: Polyhedral Patterns. *ACM Transactions on Graphics 34*, 6 (2015), 172:1–172:12. 8

[LL01] LEE S. H., LEE K.: Partial Entity Structure: a Compact Non-manifold Boundary Representation based on Partial Topological Entities. In *Proceedings of the Symposium on Solid Modeling and Applications* (2001), pp. 159–170. 2, 5

[LNPT99] LOPES H., NONATO L. G., PESCO S., TAVARES G.: Dealing with Topological Singularities in Volumetric Reconstruction. In *Proceedings of the International Conference on Curves and Surfaces* (1999). 2

[MAN] *The Mangrove Topological Data Structure (Mangrove TDS) Library*. http://mangrovetds.sourceforge.net. 8, 9

[MTP*15] MARCIAS G., TAKAYAMA K., PIETRONI N., PANOZZO D., SORKINE O., PUPPO E., CIGNONI P.: Data-driven Interactive Quadrangulation. *ACM Transactions on Graphics 34*, 65 (2015). 8, 9

[Nab96] NABUTOVSKY A.: Geometry of the Space of Triangulations of a Compact Manifold. *Communications in Mathematical Physics 181* (1996), 303–330. 1, 3, 4

[OS15] OWEN S. J., SHIH R. M.: A Template-based Approach for Parallel Hexahedral Two-Refinement. In *Proceedings of the International Meshing Roundtable* (2015), pp. 31–43. 8

[PBCF93] PAOLUZZI A., BERNARDINI F., CATTANI C., FERRUCCI V.: Dimension-Independent Modeling with Simplicial Complexes. *ACM Transactions on Graphics 12*, 1 (1993), 56–102. 2

[PTL04] PESCO S., TAVARES G., LOPES H.: A Stratification Approach for Modeling Two-dimensional Cell Complexes. *Computer & Graphics 28* (2004), 235–247. 2

[RC99] ROSSIGNAC J., CARDOZE D.: MatchMaker: Manifold BReps for Non-manifold R-sets. In *Proceedings of the Symposium on Solid Modeling and Applications* (1999), pp. 31–41. 2

[Sha08] SHAMIR A.: A Survey on Mesh Segmentation Techniques. *Computer Graphics Forum 27*, 6 (2008), 1539–1556. 1

[Si15] SI H.: TetGen: a Delaunay-based Quality Tetrahedral Mesh Generator. *ACM Transactions on Mathematical Software 41*, 2 (2015). 9

[TR80] TILOVE R. B., REQUICHA A. A. G.: Closure of Boolean Operations on Geometric Entities. *Computer-Aided Design* (1980). 2

[Whi65] WHITNEY H.: Local Properties of Analytic Varieties. In *Differential and Combinatorial Topology*. 1965, pp. 205–244. 2

[WZK16] WANG W., ZANNI C., KOBBELT L.: Improved Surface Quality in 3D Printing by Optimizing the Printing Direction. *Computer Graphics Forum 35*, 2 (2016), 59–70. 9