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 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 [BCMA11], but their efficient manipulation is challenging. Many geometry processing applications are defined on the manifolds [BKP10], 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, BKP10]. 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 desired 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 Hausdorff 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 & Stitching 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 $\mathbb{R}^3$, 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 $\Gamma$. In fact, many structural models are static (once created), and are not designed for being automatically updated, if the corresponding shape $\Gamma$ is modified. A trivial solution consists of applying the update $u$ of interest on $\Gamma$, and to obtain a new shape $\Gamma_u$. At this point, it is sufficient to recompute the decomposition $M_\Gamma$ of the new shape $\Gamma_u$ without considering the existing decomposition $M_\Gamma$ of $\Gamma$. 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 on a non-manifold shape. Thus, the naive approach is always computable, but it may be inefficient. In fact, an update $u$ may modify only a small portion of $\Gamma$, and this means that only the corresponding components in $M_\Gamma$ (modified by $u$) may be updated. Instead, the remaining parts of $\Gamma$ (not modified by $u$) are unchanged. Thus, the corresponding components in $M_\Gamma$ may be reused in $M_{\Gamma_u}$. 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 $M_\Gamma$ 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 $\gamma$, embedded in the Euclidean space $\mathbb{R}^m$, is the homeomorphic image of $B^k = \{x \in \mathbb{R}^k : ||x|| \leq 1\}$ with $0 \leq k \leq n$. Here, $k = \dim(\gamma)$ is the dimension of the cell $\gamma$. 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 $\Gamma$ contains several cells of dimension up to $d$ (and not necessarily of the same dimension) with disjoint interiors, such that the boundary $\partial \Gamma$ of every $k$-cell $\gamma$, with $0 \leq k \leq d$, belongs to the union of the cells in $\Gamma$ of dimension less than or equal to $(k-1)$. The maximum dimension $d$ of its cells is the dimension of $\Gamma$. A $d$-cell in $\Gamma$ is a maximal cell. We indicate the collection of all $k$-cells in $\Gamma$ as $\Gamma^k$ (for $0 \leq k \leq d$). A $m$-cell $\gamma$ in the $d$-complex $\Gamma$, such that $\gamma' \in B(\gamma)$, for a $k$-cell $\gamma$ in $\Gamma$ (for $0 \leq m \leq k < d$), is a $m$-face of $\gamma$. All $(k-1)$-faces in $B(\gamma)$ form the immediate boundary $\partial \gamma$ of $\gamma$. Similarly, a $m$-cell $\gamma'$, such that $\gamma \in B(\gamma')$ (with $0 \leq k < m \leq d$), is a $m$-coface of $\gamma$. The star $\text{St}(\gamma)$ contains all $m$-cofacies of $\gamma$, for all $0 \leq k < m$. All $(k+1)$-cofacies in $\text{St}(\gamma)$ forms the immediate star of the $k$-cell $\gamma$. A $k$-cell $\gamma$, such that $\text{St}(\gamma) = \emptyset$, is a top $k$-cell in $\Gamma$ (with $0 \leq k \leq d$). We indicate the collection of all top $k$-cells in $\Gamma$ as $\Gamma^k$. The top cells are not necessarily maximal cells. A $d$-complex $\Gamma$ is pure, if and only if $\Gamma^m \equiv \emptyset \neq \emptyset$, and $\Gamma^0 = \emptyset$, for all $0 \leq k \leq d$. Thus, all top $d$-cells in $\Gamma$ are maximal cells. Instead, a not pure cell $d$-complex $\Gamma$ contains several top cells, that are not maximal cells. Any two cells $\gamma_1$ and $\gamma_2$ (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 $\gamma_1$ and $\gamma_2$ (with $k \neq 0$) are adjacent, if they share a $(k-1)$-face. Instead, two adjacent vertices are connected by an edge. The carrier $\text{carrier}(\Gamma)$ of $\Gamma$ is the subset of the Euclidean space, which is spanned by the union of its Euclidean cells. Following [Art10], a digital object $O$ is described by the pair $O = (\text{carrier}(\Gamma), \Delta(\Gamma))$ in order to separate the geometric aspects in $\Delta(\Gamma)$ from the connectivity aspects in $\Gamma$.

### Combinatorial Characterization of Non-Manifolds.

It is possible to characterize a $k$-cell $\gamma$ as a (combinatorial) manifold cell in the $d$-complex $\Gamma$ by analyzing its link [DFMM03]. The link $\text{Lk}(\gamma)$ of a $k$-cell $\gamma$ consists of all cells, not in $\gamma$, but belonging to the boundary of some cells in $\text{St}(\gamma)$. All edges (in red) in Figure 1(a) belong to the link $\text{Lk}(e_2)$ for the edge $e_2$. A $d$-complex is a (combinatorial) manifold $d$-complex, if $\text{Lk}(v)$, for every vertex $v$, is homeomorphic to the tessellation of either $S^{d-1} = \{x \in \mathbb{R}^d : ||x|| = 1\}$, or to $D^{d-1} = \{x \in \mathbb{R}^d : ||x|| < 1\}$, as shown in [Ago05]. A vertex $v$, at which this condition is violated, like the vertex $v$ in Figure 1(b), is indicated as non-manifold singularities, and the $d$-complex $\Gamma$ is a non-manifold $d$-complex. Intuitively, a $k$-cell $\gamma$ in the $d$-complex $\Gamma$ is manifold, if $\text{Lk}(\gamma)$ is homeomorphic to the tessellation of either $S^k$ or to $D^{k-1}$, for any $h \leq d - k - 1$. This implies that a $(d-1)$-cell $\gamma$ is a manifold cell in the $d$-complex $\Gamma$, if $\text{St}(\gamma)$ 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 [Nab06]. 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 $\Sigma$ is a cell $d$-complex $\Sigma$, 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 $\Sigma$.

### The Manifold-Connected (MC-) connectivity.

A $k$-path $\gamma_1 \rightarrow \gamma_k$ is a sequence $(\gamma_i)_{i=1}^{k}$ of cells in a cell $d$-complex $\Gamma$ (with $0 \leq k < d$), such that the cells $\gamma_i$ and $\gamma_{i+1}$ share a $k$-face $\tau_i$. In the remainder of this paper, $\gamma_i \in (\gamma_i \rightarrow \gamma_k)$ and $\tau_i \in (\gamma_i \rightarrow \gamma_k)$ indicate that the cells $\gamma_i$ and $\tau_i$ are traversed by (or belong to) the $k$-path $\gamma_1 \rightarrow \gamma_k$. A $d$-complex $\Gamma$, 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 $\Gamma^k$ (with $1 \leq k \leq d$), have a key role in the applications. Specifically, two top $k$-cells $\gamma$ and $\gamma'$ are Manifold-Connected (MC-) adjacent, if they are $(k-1)$-adjacent along a common $(k-1)$-face $\tau$, and they are the unique top cells in $\text{St}(\tau)$. 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 $\text{MC}\equiv \subseteq \Delta(\Gamma)^2 \times \Delta(\Gamma)^2$ for two top $k$-cells $\gamma$ and $\gamma'$, such that $\gamma \stackrel{MC}{\rightarrow} \gamma'$ if:

$$\exists \tau \in \Delta^{k-1}(\text{St}(\gamma) \cap \text{St}(\gamma')) \land (S^i(\tau) = (\gamma, \gamma'))$$

Specifically, we say that a MC-adjacency occurs along the common face $\tau$. 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 $q_2$ and $q_3$ are not MC-adjacent. In any case, they belong to the 1-path $(q_1, q_0)$, 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 $\gamma_1 \rightarrow_{MC} \gamma_k$ is a $(k-1)$-path $(\gamma_i)_{i=1}^{k}$ in the $d$-complex $\Gamma$ (for any $1 \leq k \leq d$), such that all pairs of the consecutive top $k$-cells $\gamma_i$ and $\gamma_{i+1}$ are MC-adjacent. A MC-path is the transitive closure of the MC-adjacency relation on the top $k$-cells in $\Gamma$. 
Let $\gamma$ be a top $k$-cell in a cell $d$-complex $\Gamma$ (with $1 \leq k \leq d$). A Manifold-Connected (MC-) component $\Gamma_1^k$ of dimension $k$ is the collection of the top $k$-cells in $\Gamma$, that belong to every MC-path, traversing the representative cell $\gamma'$, i.e., all MC-paths of dimension $k$ in $\Gamma$, outgoing from $\gamma$. Formally, a MC-component is described by the MC-connectivity relation $\sim_{MC}^k \subseteq \Gamma_1^k \times \Gamma_1^k$, which relates a top $k$-cell $\gamma$ with any top $k$-cell $\gamma'$, reachable through a MC-path, outgoing from $\gamma$. Here, $\gamma \sim_{MC}^k \gamma'$ if:

1. $\exists \gamma'' \in \Gamma_1^k, \gamma' \in \left(\gamma \sim_{MC}^k \gamma''\right)$
2. $\exists \sigma \in \Gamma_1^k, (\sigma \neq \gamma) \wedge (\gamma' \in \Gamma_1^k)$

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

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

**Proof.** By construction, a cell $\gamma$ is MC-adjacent to itself, thus $\sim_{MC}^k$ is reflexive. Similarly, two top $k$-cells are mutually MC-adjacent, hence, $\sim_{MC}^k$ is also symmetric. Let $\gamma_1 \sim_{MC}^k \gamma$ and $\gamma \sim_{MC}^k \gamma_2$ be two MC-paths in $\Gamma_1^k$. For each $i = 1, 2$, there exists an immediate face $\tau_i$ of $\gamma$, which is traversed by $\gamma \sim_{MC}^k \gamma_i$. This implies that there exists a top $k$-cell $\gamma_i'$, such that $\gamma' \sim_{MC}^k \gamma_i'$ and $\gamma_i' \sim_{MC}^k \gamma_i$ along $\tau_i$. In other words, for each $i = 1, 2$, a MC-adjacency occurs along $\tau_i$, connecting $\gamma$ and $\gamma_i'$, and there exists a MC-path from $\gamma_i'$ to $\gamma_i$. Thus, there exists a MC-path $\gamma_1 \sim_{MC}^k \gamma_2$. This shows that $\sim_{MC}^k$ is transitive, thus it is an equivalence relation. 

This result allows to define the MC-component $\Gamma_1^k$ as the equivalence class $[\gamma] = \{ \gamma' \in \Gamma_1^k | \gamma \sim_{MC}^k \gamma' \}$, for $k \geq 1$. Its representative top $k$-cell $\gamma$ is arbitrarily chosen among the top $k$-cells, that are equivalent to $\gamma$ with respect to the MC-connectivity relation. In the remainder, we will omit the dimension $k$ in $\sim_{MC}^k$ (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 $\Gamma$, formed by only one MC-component $[\gamma]$, is a MC-complex of dimension $d$. By definition, the star of every $(d-1)$-cell $\tau$ contains one or two top $d$-cells, i.e., $\tau$ is manifold [DFMM03]. 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 $\Gamma/\sim_{MC}$ of a pure $d$-complex $\Gamma$ forms its Manifold-Connected (MC-) decomposition $MC_{\Gamma}$. The equivalence classes correspond to the MC-components of $\Gamma$. 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 $\Gamma$, containing a sparse distribution of the top $k$-cells, for every $0 \leq k \leq d$. First, the complex $\Gamma$ is decomposed into the pure subcomplexes $\Gamma_i$, formed by all top $k$-cells in $\Gamma$. Then, for $k \geq 1$, the MC-decomposition $MC_{\Gamma_i}$ is computed for every $\Gamma_i$. The $MC_{\Gamma}$ decomposition intersects the $d$-complex $\Gamma$ consists of the top vertices in $\Gamma$, and of the union of all quotient spaces $\Gamma_i/\sim_{MC}$ for every $1 \leq k \leq d$. By construction, $MC_{\Gamma}$ is unique, since every top $k$-cell $\gamma$ belongs to only one $\Gamma_i$, thus to only one MC-component.

**The Characterization of the Non-Manifold Singularities.** For the sake of simplicity, we say that the MC-component $[\gamma]$ belongs to the star $St(\tau)$ of any face $\tau$, bounding $\gamma$. Thus, the number $n_{MC}$ of MC-components in $St(\tau)$ helps to recognize efficiently the non-manifold singularities [Can12]. Specifically, if $n_{MC} > 2$, then $\tau$ is non-manifold, as well as if $St(\tau)$ contains several MC-components of different dimension. Instead, the star of a manifold $k$-cell $\tau$ contains only two $MC_{MC}$-components of dimension $k + 1$. In any case, if $n_{MC} = 1$, then $\tau$ may be either manifold, or a pinced non-manifold singularity occurs at $\tau$. In this latter case, face $\tau$ does not play a relevant role in the applications, and it can be discarded [BCMA11].

**Encoding the MC-decomposition.** The $MC_{\Gamma}$ decomposition is a $d$-complex $\Gamma$ is described by the two-level hypergraph $G_{MC}^{\Gamma}$ [CDF13]. The lower level in $G_{MC}^{\Gamma}$ contains a topological data structure $MC_{\Gamma}$, able to represent the non-manifold complex $\Gamma$. The Generalized Indexed data structure with Adjacencies [CDF1] 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 $\tau$, and the retrieval of $n_{MC}$. Instead, the upper level of $G_{MC}^{\Gamma}$ contains the MC-components in $MC_{\Gamma}$. The layers of $G_{MC}^{\Gamma}$ are independent, and decouple the local connectivity for the cells in $\Gamma$ from the structural aspects in $MC_{\Gamma}$. 

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 $\sim_{MC}^2$ 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.

© 2016 The Author(s)

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 [DFPM07], an update \( u = (u^+, u^-) \), applied on a complex \( \Gamma \), removes the collection \( u^- \) of cells from \( \Gamma \), and replaces them with the collection \( u^+ \) of new cells, that are added to \( \Gamma \). 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 \( \mathcal{MC}_\Gamma \) from \( \mathcal{MC}_\Gamma \) without its complete recomputation (the naïve approach). This is done in two steps. First, we identify what MC-components of \( \mathcal{MC}_\Gamma \), are not affected (modified) by \( u \), since they are reused directly in \( \mathcal{MC}_\Gamma \). Then, the MC-components, affected by \( u \), are modified, and saved in \( \mathcal{MC}_\Gamma \).

Formally, we say that a MC-component \( \gamma \) is affected by the update \( u = (u^+, u^-) \), if at least one top cell \( v \) in \( \gamma \) belongs to the generalized neighborhood \( \sigma^0(u^-) \), 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 \( h \neq 0 \), \( \sigma^0(u^-) \) contains all top cells in the star of the vertices, bounding the top cells in \( \sigma^{h-1}(u^-) \).

We consider the minimum order \( h \), such that \( \gamma \) intersects \( \sigma^0(u^-) \) 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 \( \sigma^0(u^-) \). For instance, \( \sigma^0(v_{01}) = \{ q_1, q_2 \} \) and \( \sigma^0(v_{00}) = \{ q_0, q_1, q_2, q_3, q_4 \} \) 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 \( \Gamma \) 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 \( e_2 \) and by the top edge \( e_2 \). The complex \( \Gamma \) 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>
<thead>
<tr>
<th>Operation</th>
<th>( V )</th>
<th>( E )</th>
<th>( F )</th>
<th>( R )</th>
<th>( L )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Make-Vertex-Region (MVR)</td>
<td>+1</td>
<td>-</td>
<td>-</td>
<td>+1</td>
<td>-</td>
</tr>
<tr>
<td>Kill-Vertex-Region (KVR)</td>
<td>-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Make-Edge-Loop (MEL)</td>
<td>-</td>
<td>+1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kill-Edge-Loop (KEL)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Make-Edge-Join-Region (MEJR)</td>
<td>-</td>
<td>+1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kill-Edge-Split-Region (KEJSR)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Make-Edge-Vertex (MEV)</td>
<td>+1</td>
<td>+1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kill-Edge-Vertex (KEV)</td>
<td>-1</td>
<td>-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Make-Face-Kill-Loop (MKFL)</td>
<td>-</td>
<td>-</td>
<td>+1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Kill-Face-Make-Loop (KFML)</td>
<td>-</td>
<td>-</td>
<td>-1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Split-Edge-Make-Vertex (SEMV)</td>
<td>+1</td>
<td>+1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Join-Edge-Kill-Vertex (JEKV)</td>
<td>-1</td>
<td>-1</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

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 \( \gamma \) is encoded for each MC-component \( \gamma \). The actual top cells in \( \gamma \) belong to the transitive closure of \( \sim_{\mathcal{MC}} \), 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 \( \mathcal{MC}_\Gamma \) (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 \( \gamma \), corresponding to a new top vertex \( v \), thus to a new region in \( \Gamma \). Instead, the KVR operator removes an existing top vertex \( v \) from \( \Gamma \) and the corresponding MC-component \( \gamma \) from \( \mathcal{MC}_\Gamma \). The time complexity of both these operators is constant.

The MEL operator. This operator creates a new top edge \( e \) between the vertices \( v_1 \) and \( v_0 \) in the same region of \( \Gamma \) (see Figure 3). The vertices \( v_1 \) and \( v_0 \) 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 \( \mathcal{St}(v_1) \) and \( \mathcal{St}(v_0) \) after adding \( e \). This latter belongs initially to the MC-component \([\gamma] \), which is merged and manipulated only if \( \mathcal{St}(v_1) \) and \( \mathcal{St}(v_0) \) contain either only one or two top edges, as follows:

- If \( \mathcal{St}(v_1) = \{ e \} \), then it is possible to merge \( e \) with the MC-component in \( \mathcal{St}(v_1) \) by considering also \( \mathcal{St}(v_0) \):
  1. if \( \mathcal{St}(v_0) = \{ e' \} \), then \( [\gamma] \) is merged with the MC-components in \( \mathcal{St}(v_1) \) and \( \mathcal{St}(v_0) \). If \( e' \) and \( e'' \) are MC-equivalent, then they belong to a MC-component \([\kappa] \), which is replaced by \([\kappa'] \), containing \( e \) and the edges in \([\kappa'] \). Otherwise, they belong to \([\kappa'] \) and \([\kappa''] \), respectively, that are replaced by \([\kappa'] \), containing \( e \) and the edges in \([\kappa'] \) and \([\kappa''] \). The time complexity of these operations is linear in the number of the top edges in \([\kappa'] \).
  2. If \( \mathcal{St}(v_0) = \{ e'' \} \), then a not MC-adjacency occurs at \( v_0 \), and \( \mathcal{St}(v_0) \) contains a MC-component \([\kappa''] \), which is split. In particular, a new MC-component \([\kappa'=\kappa''] \) 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 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 $v_1 \rightarrow_0 v_n$. 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 $v_n$, even if $e$ is removed (for $i = 1$ and $i = n$), i.e., if $St(v_1)$ contains at least one MC-component of dimension 2, and at least four top edges (including $e$). Otherwise $St(v_1)$ 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 $v_1 \rightarrow_{MC} v_n$, 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 $St(v_n) = \{e'', e'', e\}$, then it possible to merge together the existing MC-components $[e']$ and $[e'']$ in $St(v_n)$ in order to obtain a new MC-component $[e']$ (as in the previous case). If the edges in $[e']$ are MC-equivalent to the edges in $[e'']$, then they are merged into a new MC-component $[e']$. Otherwise $[e']$ and $[e'']$ remain independent.
        - if $St(v_n) = \{e', e\}$, then a new MC-component $[e']$ 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 $\partial_i^0(e)$, but the top edges of interest do not belong necessarily to $\partial_i^0(e)$.

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 $\Gamma$, that are merged into one connected region. Instead, the KESR operator removes an existing top edge $e = (v_1, v_2)$, connecting two regions of $\Gamma$. 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.

\begin{itemize}
  \item The KEL operator. This operator removes an existing top edge $e = (v_1, v_2)$ from a hole loop $\{e_1, \ldots, e_{n-1}, e\}$ in the 2-complex $\Gamma$. The star of every edge in the hole loop may be top, or may contain components of dimension 1 in $St$.
  \item The time complexity of these operations is constant.
\end{itemize}
If \( S_t(v) = \{ e, e' \} \), then it is not possible to merge \([e]\) with \([e']\), since a MC-adjacency does not occur at \(v\). The new MC-component \([e]\) is added to \(MC_G\), and it is necessary to check if \(e\) and \(e'\) belong to a closed MC-path, connecting \(v\) with itself without traversing \(v\). In other words, if \([e]\) is a loop, formed only by several top edges, then \([e]\) remains unchanged in \(MC_G\). Otherwise, \([e]\) is removed from \(MC_G\), and the new MC-component, containing all top edges that, are, respectively, MC-equivalent to \(e\) and \(e'\), is formed by \([e]\) and \([e']\). The time complexity of these operations is linear in the number of the top edges in \([e]\).

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

1. If \(S_t(v) = \{ e \}\), then \([e]\) is removed from \(MC_G\), since it contains no top edge. After removing \(e\), the vertex \(v\) becomes a top vertex, and a new MC-component \([e]\) is added to \(MC_G\). Hence, the time complexity of these operations is constant.

2. If \(S_t(v)\) 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 \(MC_G\).

The time complexity of these operations is constant.

3. If \(S_t(v)\) 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 \(\Gamma\) and \([e]\) from \(MC_G\). For the sake of simplicity, we assume that \(e\) and \(e'\) are in \(S_t(v)\). Hence, the MC-components \([e]\) and \([e']\) are removed from \(MC_G\), and a new MC-component \([e]_{\gamma}\) (which may be also indicated as \([e']_{\gamma}\)) is added to \(MC_G\). The new MC-component \([e]_{\gamma}\) 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 \(S_t(v)\) 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 \(MC_G\), and to add a new MC-component \([e]_{\gamma}\), 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 \(\gamma\), which fills the void, enclosed by an existing hole loop \(\{e_i\}^p_{i=1}\) in \(\Gamma\). After adding \(\gamma\), the hole loop is removed, and the star of each edge \(e_i\) contains at least \(\gamma\). Thus, all MC-components, formed by edges in the hole loop, are removed. At this point, it is necessary to check if \(\gamma\) 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 \(\gamma\).
2. If \(S_t(e_i) = \{ \gamma' \}\), then it is possible to merge \(\gamma\) with the existing MC-component \([\gamma']\) of dimension 2. The 2-cell \(\gamma\) is stored in \(s\).
3. If \(S_t(e_i)\) contains at least three top 2-cells in addition to \(\gamma\), then a not MC-adjacency occurs at \(e_i\).
4. If \(S_t(e_i) = \{ \gamma', \gamma'' \}\) (except \(\gamma\), then \(\gamma'\) and \(\gamma''\) 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 \(\gamma' - w_{\gamma'}\gamma''\) not traversing \(\gamma''\), 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 \(\gamma'' - w_{\gamma''}\gamma'\), not traversing \(\gamma'\), but passing through an edge \(e_{\gamma'}\) in the initial hole loop. This MC-path identifies a new MC-component \([\gamma'']\). If \(b_{\gamma''} = b_{\gamma''} = 1\), then the existing MC-component in \(S_t(e_i)\) 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 \(S_t(e_i)\) is merged with \(\gamma\), and \(\gamma'\) is added to \(s\).
   - If both \(b_{\gamma'}\) and \(b_{\gamma''}\) are not true, then the MC-component of interest in \(S_t(e_i)\) is removed, and replaced by \([\gamma']\) and \([\gamma'']\).
   - If only \(b_{\gamma'}\) is true, then the MC-component of interest in \(S_t(e_i)\) is removed, and split into \([\gamma']\), which is merged with \(\gamma\) (\(\gamma'\) is saved in \(s\)), and \([\gamma'']\), which is directly created in \(MC_G\).
   - If only \(b_{\gamma''}\) is true, then the MC-component of interest in \(S_t(e_i)\) is removed, and split into \([\gamma']\), which is merged with \(\gamma\) (\(\gamma''\) is saved in \(s\)), and \([\gamma'']\), which is directly created in \(MC_G\).

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.

The KFML operator. This operator removes an existing 2-cell \(\gamma\) without destroying its immediate edges \(\{e_i\}^p_{i=1}\), that form a new hole loop in \(\Gamma\). It is important to check if new MC-components of dimension 1, involving all edges in \(B(\gamma)\), that become top edges, after removing \(\gamma\), must be added to \(MC_G\). 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 \(\gamma\). We assume to store the representative 2-cells of these MC-components into a set \(s\), initially empty. The MC-components in \(S_t(e_i)\), with \(i = 1, \ldots, n\), are updated as follows (see Figure 6):

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

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 $\gamma$, that may be modified (if necessary). At this point, every 2-cell $\gamma_i$ in $s$ is visited as follows:

- if there does not exist $\gamma_i \sim_{MC}^{2} \gamma$ for any $\gamma$ in $s$, then $\gamma_i$ is removed from $s$, and a new MC-component $[\gamma_i]$ is created.
- Otherwise, there exists a MC-path, connecting $\gamma_i$ with several 2-cells in $s$. These latter and $\gamma_i$ 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 $B' (\gamma)$. Finally, we identify all MC-components of dimension 1, formed by all top edges in $q$. These new MC-components are added to $\mathcal{MC}_T$.

The time complexity of these operations is linear in $|q|$.

The SEMV and the JEKV operators. The SEMV operator creates a new vertex $v_i$ and splits a top edge $e = (v_1, v_2)$ into the top edges $e_1 = (v_1, v_i)$ and $e_2 = (v_i, v_2)$. 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 $v_i$. 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 $e_1 = (v_1, v_i)$ and $e_2 = (v_i, v_2)$ 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 $\mathcal{MC}_T$ of a 2-complex $\Gamma$.

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 $\gamma_1$ and $\gamma_2$, that are MC-adjacent along the edges $\{e_i\}_{i=1}^{m}$, into a new 2-cell $\gamma$ in $\Gamma$. This operation removes $\gamma_1$ and $\gamma_2$ 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 $\{e_i\}_{i=1}^{m}$ are deleted through the KEV operator, creating a new (larger) hole loop. This latter is the immediate boundary of the new 2-cell $\gamma$, 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, $\mathcal{MC}_T$ remains unchanged. Other updates, e.g., the cell and edge collapses, may modify $\mathcal{MC}_T$.

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 $\Gamma$. Broadly speaking, all 2-cells in $\mathcal{St}(e)$ are removed, and all cells, belonging only to $\mathcal{St}(v_1)$ or $\mathcal{St}(v_2)$ 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 $\mathcal{St}(e)$ 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 $\Gamma$. 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 $\Gamma$. 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,
necessary for computing the MC-decomposition $\mathcal{MC}_\Gamma$ after applying $m$ random updates on a 2-complex $\Gamma$. The time complexity for computing $\mathcal{MC}_\Gamma$ is linear in $s'$ [Can12]. First, we analyze the running time $\mathcal{T}_{\text{run}}^\mathcal{MC}$, 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 $\Gamma_k$ contains $s' = s_0'$ cells. Thus, $\mathcal{T}_{\text{run}}^\mathcal{MC}$ is linear in $\sum_{k=0}^{m-1} s'$. We also evaluate the running time $\mathcal{T}_{\text{run}}^\text{batch}$ of the batch approach, which is used for computing $\mathcal{MC}_\Gamma$ only after applying $m$ updates on $\Gamma$. Finally, we evaluate the total running time $\mathcal{T}_{\text{run}}^\text{total}$, needed for applying interactively $m$ random updates on $\Gamma$. 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 $s'$ 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 $\mathcal{T}_{\text{run}}^\mathcal{MC}$, $\mathcal{T}_{\text{run}}^\text{batch}$, and $\mathcal{T}_{\text{run}}^\text{total}$ (in milliseconds) by varying $m$. The running times, requiring more than 10 minutes of computations ($6 \times 10^3$ ms), are discarded. It is clear that the naive approach may be even feasible, even if a small number of $m$ of updates (about 40K). Already in this case, $\mathcal{T}_{\text{run}}^\mathcal{MC}$ exceeds 10 minutes of computation. Our interactive approach exhibits fast running times $\mathcal{T}_{\text{run}}^\text{batch}$ (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 [HDFO] 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,MTP15]. Our approach also allows 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

---

**Table 2:** The experimental results show the validity of our interactive approach (see the running time $\mathcal{T}_{\text{run}}^\mathcal{MC}$), after applying $m$ random updates on the MC-decompositions (with $n^\mathcal{MC}$ MC-components) for several 2-complexes with $s_0'$ cells (initially $s_0' = s_0'$), with respect to the running times $\mathcal{T}_{\text{run}}^\text{batch}$ and $\mathcal{T}_{\text{run}}^\text{total}$ 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.

<table>
<thead>
<tr>
<th>Model</th>
<th>$s'$</th>
<th>$n^\mathcal{MC}$</th>
<th>$m$</th>
<th>$s_0'$</th>
<th>$\mathcal{T}_{\text{run}}^\mathcal{MC}$</th>
<th>$\mathcal{T}_{\text{run}}^\text{batch}$</th>
<th>$\mathcal{T}_{\text{run}}^\text{total}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cubes</td>
<td>40.4K</td>
<td>4</td>
<td>1K</td>
<td>44.4K</td>
<td>19.1K 19 0.55</td>
<td>100K 440.4K 189 51</td>
<td>1.5M 6M 4.39K 823</td>
</tr>
<tr>
<td>Frame</td>
<td>40.7K</td>
<td>132</td>
<td>1K</td>
<td>44.7K</td>
<td>19.9K 21 0.58</td>
<td>100K 440.7K 196 58</td>
<td>1.5M 6.1M 4.8K 830</td>
</tr>
<tr>
<td>Twist</td>
<td>82.4K</td>
<td>4</td>
<td>5K</td>
<td>92.4K</td>
<td>183K 41 2.8</td>
<td>200K 482.4K 842 107</td>
<td>5M 6M 10.8K 1.67K</td>
</tr>
<tr>
<td>Mecha</td>
<td>86.7K</td>
<td>45</td>
<td>5K</td>
<td>96.7K</td>
<td>195K 42 2.85</td>
<td>200K 486.7K 843 107</td>
<td>5M 6.1M 10.9K 1.68K</td>
</tr>
</tbody>
</table>

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


