Homology preserving simplification for top-based representations

Federico Iuricich[1], Riccardo Fellegara[1] and Leila De Floriani[2]

[1]Department of Computer Science and UMIACS, University of Maryland, College Park (MD), USA.
[2]Department of Computer Science, Bioengineering, Robotics, and Systems Engineering, University of Genova, Genova, Italy.

Abstract

Topological features provide global information about a shape, such as the number of the connected components, and the number of holes and tunnels. These are especially important in high-dimensional data analysis, where pure geometric tools are usually not sufficient. When dealing with simplicial homology, the size of the simplicial complex Σ is a major concern, since all the algorithms available in the literature are mainly affected by the number of simplices of Σ. Edge contraction has been the most common operator for simplifying simplicial complexes. It has been used in computer graphics and visualization and more recently in topological data analysis. Edge contraction on its own does not preserve the homological information but a check, called link condition [2], has been developed for verifying whether the contraction of an edge preserves homology or not. However, since the number of simplices in the link of an edge grows exponentially when the dimension of the complex increases, checking the link condition is costly. In our work, we consider the definition of an homology preserving simplification algorithm, introducing a new way for verifying the link condition. We focus on a specific class of representations for simplicial complexes that we call top-based. A *top-based* representation encodes only the vertices and top simplices (also called *facets*) of a simplicial complex Σ, thus providing a data structure scalable with the dimension and size of Σ.

Background notions Given a *simplex of dimension p* (briefly a *p-simplex*), any simplex σ which is the convex hull of a non-empty subset of the points generating τ is called a *face* of τ. Conversely, τ is called a *coface* of σ. Given a p-simplex σ, the set of simplices for which σ is a face is called star of σ (also denoted $St(\sigma)$). If $St(\sigma) = \emptyset$, σ is called *top simplex* (or *facet*). A *simplicial complex* Σ is a finite set of simplices, such that each face of a simplex in Σ belongs to Σ, and each non-empty intersection of any two simplices in Σ is a face of both. We say that Σ is a d-simplicial complex if the largest dimension of its simplices is d.

Let Σ a d-simplicial complex, an *edge contraction* acts on Σ by contracting an edge $\varepsilon = (v_1, v_2)$ to one of its vertices (i.e. v_1). As a result, all the simplices in $St(\varepsilon)$ are removed from Σ and all the simplices in $St(v_1) - St(\varepsilon)$ are mapped into $St(v_2)$ in such a way that, for each simplex $\sigma \in St(v_1) - St(\varepsilon)$, $\mu(\sigma) = (\sigma - v_1) \cup v_2$. Thus, edge contraction is an operation linear in the number of simplices in $St(v_1)$.

The *link* of a simplex $\sigma \in \Sigma$, denoted as $Lk(\sigma)$, is the set of faces of $St(\sigma)$ that do not intersect σ. An edge $\varepsilon = (v_1, v_2) \in \Sigma$ satisfies the *link condition* if and only if $Lk(v_1) \cap Lk(v_2) = Lk(\varepsilon)$. For reducing the computational cost of extracting the links all at once, a *weaker* condition, called p-link condition, has been introduced in [3]. An edge $\varepsilon = (v_1, v_2)$ satisfies the *p-link condition* if and only if either $p \leq 0$ or $p > 0$ and every $(p-1)$-simplex $\in Lk(v_1) \cap Lk(v_2)$ is also in $Lk(\varepsilon)$. Thus, an edge $\varepsilon = (v_1, v_2)$ satisfies the link condition if and only if it satisfies the p-link condition for all $p \leq d$. Despite the fundamental reduction in the computational cost, the numerosity of p-simplices in the link of two vertices, still, can be huge depending on the dimension d of the complex. We consider solving this problem, by adapting the edge contraction and the link condition to perform on a top-based representation.

Top-based homology preserving edge contraction Encoding only the top-simplices and the vertices we can perform an edge contraction $\varepsilon = (v_1, v_2)$ by focusing on the simplices $St_{top}(\varepsilon)$, i.e. the set of top-simplices incident in the edge removed, and $St_{top}(v_1)$, i.e. the set of top-simplices incident in the vertex removed with ε. The key point is to modify the set of simplices maintaining the top-based representation valid. We recall that, in a top-based representation, each simplex $\sigma \in \Sigma$ is encoded if and only if σ is a vertex or a top simplex. Thus, while removing the set of top simplices incident in ε, it is crucial to recognize if new top simplices must be introduced.

Algorithm 1 describes the procedure that can be implemented on a top-based representation for performing an edge contraction. Each top p-simplex $\sigma \in St_{top}(\varepsilon)$ is removed with the edge (rows 5 to 10). By definition of edge contraction, all the faces of σ are removed with the exception of the $(p-1)$-faces $\gamma_1 = (\sigma - v_2)$ and $\gamma_2 = (\sigma - v_1)$, which will be merged in a single $(p-1)$-face (for example γ_2). Let $S = St_{top}(\gamma_1) \cup St_{top}(\gamma_2)$ be the set of top simplices in the star of either γ_1 or γ_2. S cannot be empty before the edge contraction, as, at least σ belongs to S. By merging γ_1 and γ_2 while removing σ, the star of the new simplex will be $St(\gamma_2) = S - \sigma$. Then, γ_2 is a new top simplex if and only if $St(\gamma_2) = \emptyset$. Notice that this condition can be verified before

Algorithm 1 contractEdge(ε,Σ)

1: **Input:** Σ is a simplicial complex
2: **Input:** $\varepsilon = (v_1, v_1)$ edge to be contracted
3: **Output:** Σ' is a simplified simplicial complex
4: // *For each top simplex in the star of ε*
5: **for each** σ in $St_{top}(\varepsilon)$ **do**
6:     $\gamma_1 = (\sigma - v_2)$
7:     $\gamma_2 = (\sigma - v_1)$
8:     **if** $St_{top}(\gamma_1) \cup St_{top}(\gamma_2) = \sigma$ **then**
9:         addTop(γ_2,Σ)
10:     removeTop(σ,Σ)
11: // *For each top simplex in the star of v_1*
12: **for each** σ in $St_{top}(v_1)$ **do**
13:     $\sigma = (\sigma - v_1) \cup v_2$
14: removeVertex(v_1)

performing the edge contraction by checking if $S = \sigma$ (rows 8 to 9). Then, working on the set of top simplices incident in v_1 (rows 12 to 13), we update $St_{top}(v_1)$, replacing v_1 with v_2, without modifying the star of any other simplex. Finally, we remove v_1 from Σ (row 14). The edge contraction becames here aan operation linear in the number of simplices in $St_{top}(v_1)$.

The link condition can be efficiently verified exploiting the top-based representation as well. From the definition of link, we can trivially say that $Lk(\varepsilon) \subseteq \{Lk(v_1) \cap Lk(v_2)\}$ thus, the link condition is satisfied when also the opposite is true. Notice that, the link of a simplex is a simplicial complex and, thus, also the intersection of two simplicial complexes is still a simplicial complex. So, we can conclude that the link condition is satisfied if the top simplices in $L = Lk(v_1) \cap Lk(v_2)$ are also in $Lk(\varepsilon)$. Computing the top simplices of L is much faster than computing the links, but, still, it is an expensive operation. Let T_s the set of simplices obtained by pairwise intersecting the simplices in $St_{top}(v_1)$ and $St_{top}(v_2)$. The top simplices of L would be obtained by removing from T_s those simplices that are not maximal in L. However, to improve scalability, we avoid storing T_s thus considering all the simplices originated by the intersection. The space complexity of the top-based approach is then $O(|T_1|+|T_2|)$ since we only need to store the top simplices incident in v_1 and v_2 while the time complexity is $O(|T_1||T_2|)$, thus, depending on the size of T_1 and T_2. In practice, it is computationally faster than the traditional (*weak*) link condition since it avoids: (i) the extraction of the faces of the simplices $\sigma \in T_1, T_2$ ($2^d - 1$ faces for each simplex); (ii) the intersection of the two sets $Lk(v_1)$ and $Lk(v_2)$; (iii) and the comparison of the resulting set with $Lk(\varepsilon)$.

Experimental results In our evaluation, we use eleven simplicial complexes having from 9 thousand to 14 millions vertices. The dimension of the top simplices goes from 7 to 68. The hardware configuration used for these experiments is an Intel i7 3930K CPU at 3.20Ghz with 64 GB of RAM. We have implemented the simplification approach, using a specific top-based representation, the *Stellar tree* [4]. On the top of it we have implemented both the *weak* link condition [2] and the new *top-based* approach for verifying the link condition. The size of each simplicial complex Σ is then reduced applying homology preserving edge contractions until no more can be applied without changing the homology of Σ. A simplification process is killed after 25 hours of computation. The simplification ratio is, on average, around 90%-95% of the initial number of vertices. From the results obtained the limitation of the approach based on the *weak* link condition is evident. Using the latter, the simplification process ends in very few cases (two datasets). Typically the computation exceeds the 25 hours and, in two cases, the process exceeds the 64GB of memory. Verified the computational improvement for checking the link condition we have evaluated the practical relevance of the proposed approach comparing the performances of our implementation with respect to the state-of-the-art data structure for performing edge contractions, the *Skeleton-Blocker* [2] (as implemented in [1]). From the results obtained we can say that the Stellar tree is generally faster taking 25% to 70% of the time required by the Skeleton-Blocker for simplifying a dataset. Focusing on the timings distribution, the Skeleton-Blocker needs more effort for updating the structure during the edge contraction while it is particularly fast at verifying the link condition. Conversely, the Stellar tree is faster at performing the contraction but it is slower at checking the link condition. This is an expected result due to the characteristics of the two data structures (details are are not included for brevity). Analyzing the memory peak, i.e. the amount of memory used at runtime for performing simplification, the Stellar tree is also more compact, using from 30% to 80% of memory required by Skeleton-Blocker.

References

[1] Geometric Understanding in Higher Dimensions (GUDHI), 2016.

[2] D. Attali, A. Lieutier, and D. Salinas. Efficient Data Structure for Representing and Simplifying Simplicial Complexes in High Dimensions. In *International Journal of Computational Geometry & Applications*, volume 22, pages 279–303, New York, New York, USA, 2012. ACM Press.

[3] T. K. Dey, A. N. Hirani, B. Krishnamoorthy, and G. Smith. Edge Contractions and Simplicial Homology. *arXiv:1304.0664*, apr 2013.

[4] R. Fellegara. *A spatio-topological approach to the representation of simplicial complexes and beyond.* PhD thesis, Department of Computer Science (DIBRIS), Universitá degli Studi di Genova, 2015.