Topology Preserving Thinning of Cell Complexes

Paweł Dłotko, and Ruben Specogna, *Member, IEEE*

***Abstract*— A topology preserving skeleton is a synthetic representation of an object that retains its topology and many of its significant morphological properties. The process of obtaining the skeleton, referred to as skeletonization or thinning, is a very active research area. It plays a central role in reducing the amount of information to be processed during image analysis and visualization, computer-aided diagnosis, or by pattern recognition algorithms. This paper introduces a novel topology preserving thinning algorithm, which removes *simple cells*—a generalization of simple points—of a given cell complex. The test for simple cells is based on *acyclicity tables* automatically produced in advance with homology computations. Using acyclicity tables render the implementation of thinning algorithms straightforward. Moreover, the fact that tables are automatically filled for all possible configurations allows to rigorously prove the generality of the algorithm and to obtain fool-proof implementations. The novel approach enables, for the first time, according to our knowledge, to thin a general unstructured simplicial complex. Acyclicity tables for cubical and simplicial complexes and an open source implementation of the thinning algorithm are provided as an additional material to allow their immediate use in the vast number of applications arising in medical imaging and beyond.**

***Index Terms*— Skeleton, skeletonization, iterative thinning, topology preservation, homology, topological image analysis.**

I. Introduction

THINNING (or skeletonization) is the process of reducing an object to its skeleton. The topology preserving skeleton may be informally defined as a thinned subset of the object that retains the same topology of the original object and often many of its significant morphological properties. Thinning is a very active research area thanks to its ability of reducing the amount of information to be processed for example in medical image analysis and visualization as well as simplifying the development of pattern recognition or computer-aided diagnosis algorithms. Hence, it is not surprising that thinning gained a pivotal role in a wide range of applications. An exhaustive review of the literature is beyond the scope of this work. 2D skeletons have been used for digital image analysis and processing, optical character and fingerprint recognition, binary image compression, pattern recognition and matching since a long time ago. See for example the survey paper [1]. More recently, 3D skeletons have been widely used in computer vision and shape analysis [2], in computer graphics for mesh animation [3] and in computer aided design (CAD) for model analysis and simplification [4], [5] and for topology repair [6].

There is also a vast literature of applications of skeletons in medical imaging. They have been used for route planning in virtual endoscopic navigation [7], for example in virtual colonoscopy [8]–[12] or bronchoscopy [13]. Skeletons have also been an important part of clinical image analysis by providing centerlines of tubular structures. In particular, there is a large body of literature showing applications of skeletons to blood veins centerline extraction from angiographic images [12], [14]–[20], and intrathoracic airway trees classification [21]. Also protein backbone models are produced with skeletons [22]. Furthermore, many computer-aided diagnostic tools rely on skeletons. For example, skeletons have been used to identify blood vessels stenoses [23]–[25], tracheal stenoses [26], polyps and cancer in colon [27] and left atrium fibrosis [28].

There are medical applications of skeletons where topology preservation is essential. Non invasively determine the 3D topological network of the trabecular bone [29] is a good example. Indeed, many studies demonstrate that the elastic modulus and strength of the bones is determined by the topological interconnections of the bone structure rather than the bone volume fraction [30], [31]. Therefore, topological analysis plays a fundamental role in computer-aided diagnostic tools for osteoporosis [30].

Topology preserving thinning is non trivial and a vast literature, briefly surveyed in Section I-A, has been dedicated to this topic. In particular, thinning by iteratively removing *simple points* [32] is a widely used and effective technique. It works locally, therefore it is efficient and easy to implement. While reading the literature one may notice that thinning algorithms are claimed to be "topology preserving," even though in most cases a precise statement of what that means is left unaddressed. This paper uses *homology theory* [33] to rigorously define what the virtue of being topology preserving actually consists of. This theory is less intuitive than the concepts used so far, including simple homotopy type [34], but exhibits some important theoretical and practical advantages that will be highlighted later in the paper. We remark that a homological definition of simple points has already been used in the context of skeletonization in [16] and [35], but only in the case of cubical complexes.

Manuscript received March 12, 2014; revised June 4, 2014; accepted August 11, 2014. Date of publication August 15, 2014; date of current version September 9, 2014. The work of P. Dłotko was supported in part by the Defense Advanced Research Projects Agency, Arlington, VA, USA, under Grant FA-9550-12-1-0416, and in part by the Air Force Office of Scientific Research, Arlington, VA, USA, under Grant FA-9550-14-1-0012. The work of R. Specogna was supported by the Italian Ministry of Education, University and Research Project under Grant PRIN-2009LTRYRE. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Quanzheng Li.

P. Dłotko is with the Department of Mathematics, University of Pennsylvania, Philadelphia, PA 19104 USA, and also with the Institute of Computer Science, Jagiellonian University, Kraków 31-007, Poland (e-mail: dlotko@sas.upenn.edu).

R. Specogna is with the Department of Electrical, Mechanical and Management Engineering, Universitá di Udine, Udine 33100, Italy (e-mail: ruben.specogna@uniud.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2014.2348799

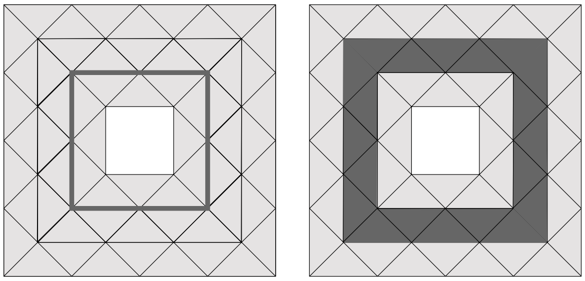

Fig. 1. Let us consider, as an example, a 2D simplicial complex $\mathcal{K}$ representing an annulus. On the left, the thick cycle represents a 1D skeleton of $\mathcal{K}$ obtained by means of standard *collapsing* of $\mathcal{K}$ [33]. On the right, the gray triangles represent a 2D skeleton of $\mathcal{K}$ according to the definition used in this paper. This kind of skeleton is obtained after removing a sequence of top dimensional cells.

This paper generalizes this idea to cell complexes that are more general than cubical complexes. There are many applications that would benefit from an algorithm for general unstructured simplicial complexes [36, p. 35]. In fact, the geometry of 3D objects is frequently specified by a triangulated surface, obtained for example by using an isosurface algorithm as *marching cubes* [36, p. 539], [12] applied on voxel data from computed tomography, magnetic resonance imaging or any other 3D imaging technique. Another possibility is to obtain the triangulations from the convex hull of point clouds provided for example by 3d laser scanners. Triangulated surfaces offer two potential advantages over voxel representation. They allow to adaptively simplify the surface triangulation, see [36, p. 549, Fig. 16.20]. They also allow to visualize and edit the object efficiently with off-the-shelf software (for example with the many tools for stereo lithography) and without the starcase artifacts typical of voxel representation of objects with curved boundary. One may even easily print the object with additive manufacturing technology (i.e. 3d printers).

Another issue that arises reading the literature is that many different definitions of topology preserving skeleton exist. In some papers, the skeleton is obtained by removing simple pairs in the spirit of simple homotopy theory by what is well known as *collapsing* in algebraic topology [33]. The resulting skeleton, if no other constraints are used, has a lower dimension with respect to the input complex. On the contrary, this paper assumes that the skeleton is always a solid object of the same dimension as the initial complex. The difference is highlighted in Fig. 1.

In this paper the skeleton of a given complex $\mathcal{K}$ is defined as a subset $S(\mathcal{K}) \subset \mathcal{K}$ that is obtained from $\mathcal{K}$ after removing a sequence of top dimensional cells. We require that the *homology* [33] of the initial complex $\mathcal{K}$ is preserved during this process. In particular, a top dimensional cell can be safely removed if this does not change the homology of the *complement* of $S(\mathcal{K})$. Fig. 2 provides an intuitive explanation why the last requirement is desirable. This additional requirement, to the best of our knowledge, is not documented in other papers. We call this cell a *simple cell*, which is a generalization of the idea of *simple points* [32] in digital topology. Clearly, in nontrivial cases, the skeleton $S(\mathcal{K})$ is not unique.

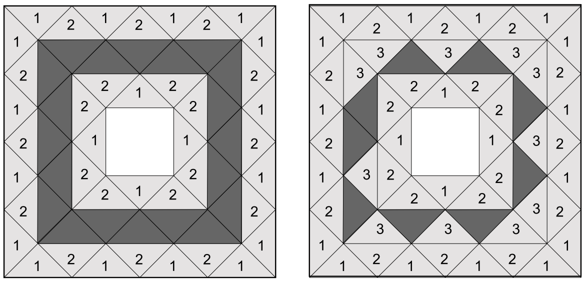

Fig. 2. Suppose the iterative thinning Algorithm 1 is used to skeletonize a 2D simplicial complex $\mathcal{K}$ representing an annulus. Let the dark gray triangles belong to the skeleton $S(\mathcal{K})$. On the left, the result obtained by checking whether the removal of a cell changes the topology of $S(\mathcal{K})$ complement. On the right, the result obtained by checking whether the removal of a cell changes the topology of $S(\mathcal{K})$. The numbers inside triangles indicate the iteration number of the while loop in the thinning algorithm when they were removed. Both skeletons preserve topology. However, in most applications, the skeleton on the left is preferred.

Resorting to explicit homology computations to detect simple points as in [16], [32], and [35] is quite computationally intensive, as the worst-case complexity of homology computations is cubical, see also the discussion in [32]. In this paper, we introduce a much more efficient solution by exploiting the idea of tabulated configurations, i.e. *acyclicity tables*, that are described in detail in Section III.

Usually, a skeleton also requires to preserve the shape of the object. In this paper we show some very simple proof of concept idea how to preserve both homology and shape. Of course, this is just an example to illustrate how the idea of acyclicity tables can be used together with some additional techniques that guarantee shape preservation.

The paper is organized as follows. In Section I-A the prior work on thinning algorithms is surveyed. Section I-B analyzes the original contributions of the present paper. In Section II the property of being a homology preserving thinning is rigorously stated. In Section III the concept of acyclicity tables is introduced, whereas, in Section IV, the topology preserving thinning algorithm is presented. Section V discusses the results of the thinning algorithm on a number of benchmarks and, finally, in Section VI the conclusions are drawn.

A. Prior Work

There are hundreds of papers about thinning. Most of them fall into two categories. On one hand, there are papers using morphological operations like erosion and dilatations to obtains skeletons, see [37] and references therein. They do not guarantee topology preservation in general. The others use the idea of removing the simple points from the given cell complex, see [16], [32], [35]. Without pretending to be exhaustive, in the following we resume previous results.

1) 2D Images: Most of the work on thinning regard finding skeletons of 2D images. A very comprehensive survey on this topic may be found in [1]. This case is well covered in literature and general solution exists, see [38]–[43].

2) 3D Cubical Complexes: In case one wants to skeletonize three (or higher) dimensional images, there are much less papers available in literature. Most of them rely on case study, see [44]–[57]. The problem is that it is hard to prove that a rule-based algorithm is general, i.e. it removes a cell if and only if its removal does not change topology. In 3d there are more than 134 millions possible configurations for a cube neighborhood and only treating correctly *all* of them gives a correct thinning algorithm. References [16], [32], and [35] use explicit homology computations to detect simple points.

There are papers presenting thinning algorithms for 3D images in which Euler characteristic is used to guarantee topology preservation, see [43], [54] and references therein. The problem is that Euler characteristic is a rather raw measure of topology and it is not sufficient to preserve topology in general for 3D cubical complexes. For 3D images one needs to use both Euler characteristic and connectivity information to preserve topology, but this is not sufficient for 4D images.

3) 2D Simplicial Complexes and Cell Complexes: All the strategies presented so far are applicable only to cubical grids (pixels, voxels, $\cdots$). To our best knowledge, there are just a few papers dealing with 2d grids that are not cubical, and they are restricted to 2d binary images modeled by a quadratic, triangular, or hexagonal cell complex, see [58]–[61]. The main reason for the lack of results on general 2d simplicial complexes may be the absence of regularity in unstructured simplicial grids that makes case-study algorithms very hard to devise and to implement. This gap in the literature is covered by the present paper.

4) 3D Simplicial Complexes and Cell Complexes: We are not aware of algorithms that deal with unstructured 3d simplicial complexes or more general cell complexes. There are only some papers that find the 1D skeleton by using the well known collapsing in algebraic topology [62]–[65]. Again, this gap in the literature is covered by this paper.

B. Summary of Paper Contributions

In this Section the main novelties presented in this paper are summarized:

1) The claim "topology preserving thinning" is rigorously defined for any cell complex with homology theory.
2) A novel topology preserving thinning algorithm that removes simple cells is introduced. This algorithm falls into the category of thinning algorithms based on simple points and generalizes all previous papers. In fact, the acyclicity tables introduced in this paper give a classification of all possible simple points that can occur in a cell complex. Therefore, no rules are needed since all of them are encoded into acyclicity tables.
3) The most important advantage of the novel approach is that acyclicity tables are *automatically* filled in advance, for any cellular decomposition, with homology computations performed by a computer. Therefore, once the tables are available, the implementation of a thinning algorithm is straightforward since identifying simple cells requires just queering the acyclicity table. No other topological processing is needed.
4) The fact that acyclicity tables are filled *automatically* and *correctly*, for *all* possible configurations, provides a rigorous computer-assisted mathematical proof that the homology-based thinning algorithm preserves topology. It is also verified, simply by checking all acyclic configurations, that using Euler characteristic is not enough to ensure preservation of topology in 3D or higher dimensional cubical and simplicial complexes. However, when one checks both Euler characteristic and that the number of connected component before and after cell removal remains one, then topology is preserved. Checking Euler characteristic together with connectivity does not suffice in 4d.
5) The acyclicity tables for simplicial complexes of dimension 2, 3 and 4 and for cubical complexes of dimension 2 and 3, that can be freely used in any implementation of the proposed algorithm, are provided as supplemental material at [66]. This way, we dispense readers to implement homology computations.
6) The thinning algorithm, unlike the standard collapsing of algebraic topology [33], does not require the whole cell complex data structure but it uses only the top dimensional elements of the complex, with obvious memory saving.
7) As a proof of concept, an open source C++ implementation that works for 3D simplicial complexes is provided to the reader as supplemental material at [66]. We remark that the code is optimized for readability and memory usage and not for speed.

II. Topology Preserving Thinning by Preserving Homology

When one claims that an algorithm "preserves topology," in order to give a precise meaning to this statement, one needs to specify which topological invariant is preserved. In the literature, the invariant is assumed to be, in most cases implicitly, the so called homotopy type [33]. The problem of this choice is that this strong topological invariant in general is not computable according to Markov [67]. This is the reason why in this paper we propose to use homology theory—which is computable—in place of homotopy theory. Indeed, homology seems to be the strongest topological invariant that can be rigorously and efficiently computed. Therefore, every time we claim that topology is not changed, implicitly we mean that the homology is not changed.

Homology groups may be used to measure and locate holes in a given space. 0D holes are the connected components. 1D holes are handles of a given space, whereas 2D holes are voids totally surrounded by the considered space (i.e. cavities). One can look at a n-dimensional hole as something bounded by a deformed n-sphere. A space is homologically trivial (or *acyclic*) if it has one connected component and no holes of higher dimensions. A rigorous definition of homology groups is not presented in this paper due to the availability of rigorous mathematical introductions in any textbook of algebraic topology as [33] and the lack of space. For a more intuitive presentation for non mathematicians one may consult [68], [69].

In this paper, we consider in particular two standard ways of representing spaces, namely the *simplicial* and *cubical* complexes. A *n-simplex* is the convex hull of $n+1$ points in general position (point, edge, triangle, tetrahedron, 4D tetrahedron). A simplex spanned with vertices $x_1, \ldots, x_n$ is denoted by $[x_1, \ldots, x_n]$. By a *face* of a n-simplex A we mean the simplices spanned by a proper subset of vertices spanning A. A *simplicial complex* $\mathcal{S}$ is a set of simplices such that for every simplex $A \in \mathcal{S}$ and every face B of simplex A, $B \in \mathcal{S}$. Pixels (2-cubes) and voxels (3-cubes) are widely used in image analysis. They form a Cartesian grid, that is cells are unit squares or unit cubes and vertices have integer coordinates. Even though we assume to deal with a Cartesian grid, the results presented in this paper hold also for more general grids such as a rectilinear grid, that is a tessellation of the space by rectangles or parallelepipeds that are not, in general, all congruent to each other. Therefore, we define a cubical complex $\mathcal{K}$ as a set of cubes such that for every cube $A \in \mathcal{K}$ and for every B being face of we have A, $B \in \mathcal{K}$. We want to stress here that we assume every cell to be closed, i.e. if a cell is present in a complex, so do are its faces.

In the iterative thinning algorithm presented in this paper, the top dimensional cells are iteratively removed from the object. Homology theory is used to ensure that removing of a given cell does not change the topology of the object. If removal of a cell does not change the topology, the cell is said to be *simple*. Due to efficiency reasons, the homology cannot be recomputed after removing every single element. In fact, one may compute the homology of a cell complex for instance with [70] software, but the worst case computational complexity is cubical. Therefore, the main idea is to rely on the so called Mayer–Vietoris sequence [33]. Let us express the considered space X as $X = X' \cup V$, where V is a single top dimensional simplex or voxel. The sequence states that once the intersection $X' \cap V$ is homologically trivial, then the homology of X' is the same as the homology of X. This important result is the key of the method presented in this paper. In fact, it implies that, in order to preserve the homology of X with respect to X', one should check the homology of the intersection $X' \cap V$. In practice, this may be easily performed with the [70] software.

The main novelty of this paper is to present a different idea to speed up the computations. Let V be a simplex or voxel. By *bd* V we denote the boundary of V, i.e. *all* lower dimensional cells which are entirely contained in the closure of V. The idea is based on the observation that in *bd* V there are not too many elements, namely

1) 6 in case of triangle (2D simplex);
2) 14 in case of tetrahedron (3D simplex);
3) 30 in case of 4D simplex (i.e. the convex hull of 5 points in $\mathbb{R}^4$ in general position);
4) 8 in case of pixel (2D cube);
5) 26 in case of voxel (3D cube).

By *configuration* we mean any subset of *bd* V. When looking for simple cells, the configuration characterizes the way V intersects the complement of the set that we aim to thin. A configuration is *acyclic* if its homology—computed as the homology of the corresponding chain complex—is trivial. Since the number of all possible configurations in *bd* V is 2^i, where i is the number of boundary elements of V, one may pre-compute the homology of all the configurations and store them in a lookup table. In this case, homology computations are done only in a pre-processing stage and once and for all. After creating them, one may instantly (i.e. in $O(1)$ time) get the answer whether the intersection $X' \cap V$ is homologically trivial or not. This is the strategy that we aim to use in the thinning algorithm. Next section shows how to use the acyclicity tables and how to obtain them automatically.

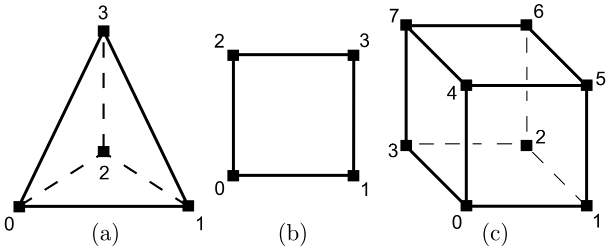

Fig. 3. The model of the (a) 3D simplex (tetrahedron of edges 01,02,03,12,13,23 and faces 012,013,023,123), the (b) 2D cube (pixel of edges 01,02,13,23) the (c) 3D cube (voxel of edges 01,03,04,12,15,23,26,37,45,47,56,67 and faces 0123,0145,0347,1256,2367,4567).

III. Use of Acyclicity Tables and Their Generation

Let us consider a generic cell V of a cell complex. Let us fix an order of all boundary elements $\{b_1, \ldots, b_n\}$ of V. We consider all subsets of the set $\{b_1, \ldots, b_n\}$ and enumerate them in the following way. For $J \subset \{1, \ldots, n\}$ the number of a subset $\{b_i\}_{i \in J}$ is $\sum_{i \in J} 2^i$. The *acyclicity table* of V is an array of size 2^n having, at position $j = \sum_{i \in J} 2^i$, the value *true* if the configuration $\{b_i\}_{i \in J}$ is acyclic and *false* otherwise.

Let us describe how the acyclicity table is constructed and used starting by considering a tetrahedron (3D simplex), see Fig. 3a. Let us enumerate vertices, edges and faces of tetrahedron as in Fig. 3a. We now introduce an ordering on boundary elements of the 3D simplex (bold numbers are the indexes of elements in the given order):

$$\begin{array}{ccccccc} \mathbf{1} & \mathbf{2} & \mathbf{3} & \mathbf{4} & \mathbf{5} & \mathbf{6} & \mathbf{7} \\ 0 & 1 & 2 & 3 & 01 & 02 & 03 \\ \mathbf{8} & \mathbf{9} & \mathbf{10} & \mathbf{11} & \mathbf{12} & \mathbf{13} & \mathbf{14} \\ 12 & 13 & 23 & 012 & 013 & 023 & 123. \end{array} \tag{1}$$

Let $l_1, \ldots, l_k$ be the indexes of elements in the considered configuration (i.e. bold numbers corresponding to elements that are present in the configuration). The index of the configuration in the acyclicity table is computed with

$$index := \sum_{i=1}^{k} 2^{l_i}.$$

The acyclicity table is generated as follows. All possible configurations of the elements are automatically generated and the homology group of each of these configurations is computed with the [70] software. If a configuration turns out to be acyclic, then a *true* is set to the place in the array corresponding to the examined configuration, *false* otherwise.

Example 1: Suppose the 3D simplex $[4, 5, 19, 20]$ *is given as input. Let* $[4, 5, 19]$ *and* $[20]$ *be the maximal elements in the configuration (i.e. the configuration consists of those elements and the vertices* $[4], [5], [19]$*, edges* $[4, 5]$, $[4, 19]$, $[5, 19]$ *that are the faces of* $[4, 5, 19]$*). This configuration needs to be mapped into the 3D simplex model presented in Fig. 3a. Hence, we have the following mapping between vertices:* $4 \to$ *0,* $5 \to$ *1,* $19 \to$ *2,* $20 \to$ *3. It is naturally extended to the mapping on simplices. Namely, the triangle* $[4, 5, 19]$ *is mapped to [0, 1, 2] in the 3D simplex model, whereas vertex* $[20]$ *is mapped to vertex [3]. Therefore, the elements in this configuration are:*

1) *Vertices: [0], [1], [2], [3] (indices 1, 2, 3, 4);*
2) *Edges: [0, 1], [0, 2], [1, 2] (indices 5,6,8);*
3) *Face: [0, 1, 2] (index 11).*

Consequently, the index of this configuration is

$$index = 2^1 + 2^2 + 2^3 + 2^4 + 2^5 + 2^6 + 2^8 + 2^{11} = 2430.$$

One may check, at this position of the provided acyclicity table for 3D simplices, that this configuration is not acyclic[1].

In the same spirit, we introduce an ordering for the 2D simplex

$$\begin{array}{cccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}&\mathbf{6}\\ \texttt{0}&\texttt{1}&\texttt{2}&\texttt{01}&\texttt{02}&\texttt{12},\end{array}\tag{2}$$

and for the 4D simplex

$$\begin{array}{cccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}&\mathbf{6}\\ \texttt{0}&\texttt{1}&\texttt{2}&\texttt{3}&\texttt{4}&\texttt{01}\\ \mathbf{7}&\mathbf{8}&\mathbf{9}&\mathbf{10}&\mathbf{11}&\mathbf{12}\\ \texttt{02}&\texttt{03}&\texttt{04}&\texttt{12}&\texttt{13}&\texttt{14}\\ \mathbf{13}&\mathbf{14}&\mathbf{15}&\mathbf{16}&\mathbf{17}&\mathbf{18}\\ \texttt{23}&\texttt{24}&\texttt{34}&\texttt{012}&\texttt{013}&\texttt{014}\\ \mathbf{19}&\mathbf{20}&\mathbf{21}&\mathbf{22}&\mathbf{23}&\mathbf{24}\\ \texttt{023}&\texttt{024}&\texttt{034}&\texttt{123}&\texttt{124}&\texttt{134}\\ \mathbf{25}&\mathbf{26}&\mathbf{27}&\mathbf{28}&\mathbf{29}&\mathbf{30}\\ \texttt{234}&\texttt{0123}&\texttt{0124}&\texttt{0134}&\texttt{0234}&\texttt{1234}.\end{array}\tag{3}$$

In the case of cubes, unlike the case of simplices, the model cube is expressly needed to specify the location of vertices in the cube[2]. The model for 2- and 3D cubes is represented in Fig.s 3b and 3c. The ordering for the 2D cube is

$$\begin{array}{cccccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}&\mathbf{6}&\mathbf{7}&\mathbf{8}\\ \texttt{0}&\texttt{1}&\texttt{2}&\texttt{3}&\texttt{01}&\texttt{02}&\texttt{13}&\texttt{23},\end{array}\tag{4}$$

whereas, for the 3D cube (voxel) is

$$\begin{array}{cccccc}\mathbf{1}&\mathbf{2}&\mathbf{3}&\mathbf{4}&\mathbf{5}&\mathbf{6}\\ \texttt{0}&\texttt{1}&\texttt{2}&\texttt{3}&\texttt{4}&\texttt{5}\\ \mathbf{7}&\mathbf{8}&\mathbf{9}&\mathbf{10}&\mathbf{11}&\mathbf{12}\\ \texttt{6}&\texttt{7}&\texttt{01}&\texttt{03}&\texttt{04}&\texttt{12}\\ \mathbf{13}&\mathbf{14}&\mathbf{15}&\mathbf{16}&\mathbf{17}&\mathbf{18}\\ \texttt{15}&\texttt{23}&\texttt{26}&\texttt{37}&\texttt{45}&\texttt{47}\\ \mathbf{19}&\mathbf{20}&\mathbf{21}&\mathbf{22}&\mathbf{23}&\mathbf{24}\\ \texttt{56}&\texttt{67}&\texttt{0123}&\texttt{0145}&\texttt{0347}&\texttt{1256}\\ \mathbf{25}&\mathbf{26}&&&&\\ \texttt{2367}&\texttt{4567}.&&&&\end{array}\tag{5}$$

Of course, in order to compute the index in the acyclicity table, exactly the same procedure as the one described for the 3D simplex is used.

Historically, acyclicitiy tables for cubes [71] and simplices [72] were introduced in order to speed up homology computations. In this paper we provide an even stronger result. Not only the homology classes of the initial set and its skeleton are the same, but one can construct a retraction from the initial set to its skeleton. The existence of retraction implies the isomorphism in homology, but the existence of retraction is a stronger property than homology preservation. We demonstrated the existence of a retraction by a brute-force computer assisted proof, i.e. by checking all acyclic configurations and constructing a retraction for each of them. Thus, the following lemma holds.

Lemma 3.1: For every acyclic configuration C *in the boundary of 2-, 3- or 4D simplices and 2- or 3D cubes (denoted as* $bd(K)$*) there exist a simple homotopy retraction from* $bd(K) \setminus C$ *to* C*.*

At the end of this section, let us define more rigorously a *simple cell.*

Definition 3.2: A cell T *in a complex* $\mathcal{K}$ *is simple if* $(bd\ T \setminus (T \cap (\mathcal{K} \setminus T)))$ *is acyclic.*

In the supplemental material, we provide the acyclicity tables for $2, 3, 4$ dimensional simplices and $2, 3$ dimensional cubes (pixels, voxels), in such a way that the reader can safely bypass the step of constructing them. We note that we do not provide tables for higher dimensional elements, since the memory required to store them is huge[3].

IV. Topology Preserving Thinning Algorithm

In this section we propose a simple thinning technique that iteratively removes simple cells. The algorithm is valid both for cubes and for simplices provided that the appropriate acyclicity table is used. We want to point out that the algorithm works on top dimensional cells (cubes, simplices). Therefore—unlike the case of homological algorithms or collapsing—there is no need to generate the whole lower dimensional cell complex data structure[4]. The input of the algorithm consists of a list $\mathcal{K}$ of top dimensional cells in the considered set. The output is a subset of $\mathcal{K}$ being its skeleton.

At the beginning, we present a first version of the algorithm that preserves only the topology of $\mathcal{K}$. One first searches the list $\mathcal{K}$ to find all the cells $K_1, \ldots, K_n$ that are simple and store them into a queue L. Then, the queue L is processed as long as it is not void. In each iteration, an element K is removed from the queue L. Then, with the acyclicity table, one has to check if K is simple in the set $S(\mathcal{K})$. We want to point out that elements already removed form the considered set $S(\mathcal{K})$ in previous iterations are treated as the exterior of $S(\mathcal{K})$ at a given iteration. If K is simple in $S(\mathcal{K})$, then it is removed from the set $S(\mathcal{K})$. In this case, all neighbors of K[5]

[1] Clearly it cannot be, since it has two connected components.

[2] This happens because in a cube not all vertices are connected with edges as in case of simplices. Therefore, the model cube is needed to point out the incidences of the vertices.

[3] All configurations for 4D cube require almost 10^9 PB. All configurations for 5D simplex require 4096 PB. On the contrary, the acyclicity table for the 3D simplices provided as supplemental material requires no more than 32kB.

[4] This structure have to be generated only locally for the boundary of a cell T when checking if T is simple.

[5] A neighbor of cell/simplex K is any cell/simplex $K_1 \in \mathcal{K}$ such that $K \cap K_1 \neq \emptyset$.

Algorithm 1 Topology Preserving Thinning

```
Require: List of maximal cells $\mathcal{K}$;
Ensure: List of maximal cells $S(\mathcal{K})$ that belong to the skeleton of $\mathcal{K}$;
Queue $L$;
$S(\mathcal{K}) = \mathcal{K}$;
for Every element $T \in S(\mathcal{K})$ do
    if $bd\ T \ \setminus (T \cap (S(\mathcal{K}) \setminus T))$ is acyclic (check with acyclic table) then
        $L.enqueue(T)$;
while $L \neq \emptyset$ do
    $T = L.dequeue()$;
    if $T \notin S(\mathcal{K})$ then
        Continue;
    if $(bd\ T \ \setminus (T \cap (S(\mathcal{K}) \setminus T)))$ is acyclic then
        $S(\mathcal{K}) = S(\mathcal{K}) \setminus T$;
        Put all neighbor cells of $T$ in $S(\mathcal{K})$ to the queue $L$;
return $S(\mathcal{K})$;
```

Algorithm 2 Shape and Topology Preserving Thinning

```
Require: List of maximal cells $\mathcal{K}$;
Ensure: List of maximal cells $S(\mathcal{K})$ that belong to the skeleton of $\mathcal{K}$;
 1: Queue $L$;
 2: $S(\mathcal{K}) = \mathcal{K}$;
 3: for Every element $T \in S(\mathcal{K})$ do
 4:     if $bd\ T \ \setminus (T \cap (S(\mathcal{K}) \setminus T))$ is acyclic (check with acyclic table) then
 5:         $L.enqueue(T)$;
 6: Queue $K$;
 7: while $L \neq \emptyset$ do
 8:     $T = L.dequeue()$;
 9:     if $T \in S(\mathcal{K})$ then
10:         if $(bd\ T \ \setminus (T \cap (S(\mathcal{K}) \setminus T)))$ is acyclic then
11:             $S(\mathcal{K}) = S(\mathcal{K}) \setminus T$;
12:         Put all the neighbor cells of $T$ in $S(\mathcal{K})$ to the queue $K$;
13:     if $L = \emptyset$ then
14:         $L = K$;
15:         $K = \emptyset$;
16:         if all the cells in $S(\mathcal{K})$ have a top dimensional face in external boundary then
17:             Break;
18: return $S(\mathcal{K})$;
```

that are still in $S(\mathcal{K})$ are added to the queue L. The details of the presented procedure are formalized in Algorithm 1. We want to stress that Algorithm 1 is just an illustration. It may be turned into an efficient implementation by using more efficient data structures (for instance removing from the list $S(\mathcal{K})$ can be replaced by a suitable marking the considered element.) Also searching for intersection of T with current $S(\mathcal{K})$ should be performed by using hash tables that, for the sake of clarity, are not used explicitly in Algorithm 1. Let us now discuss the complexity of the algorithm. Clearly the *for* loop requires $O(\|\mathcal{K}\|)$ operations. We assume that one can set and check a flag of every cell in a constant time. This flag indicates if a cell is removed from $S(\mathcal{K})$ or not. Every cell $T \in \mathcal{K}$ appears in the *while* loop only k times, where k is maximal number of neighbors of a top dimensional cell in the complex. Therefore, the *while* loop performs at most $k\,\|\mathcal{K}\|$ iterations before its termination. The time complexity of every iteration is $O(k)$, which means that the overall complexity of the procedure is $O(k^2\,\|\mathcal{K}\|)$. Typically the number k is a dimension dependent constant and, in this case, the complexity of the algorithm is $O(\|\mathcal{K}\|)$. The same complexity analysis is valid for Algorithm 2.

We now present in Algorithm 2 a simple idea that enables to preserve the shape of the object in addition to its topology. We stress that the aim of this second algorithm is just to show how to couple topology and shape preservation.

In Algorithm 2 there is one difference with respect to Algorithm 1. After removing a single external layer of cells, a check is made at line 2 to determine whether all cells that remain in $S(\mathcal{K})$ are already in the boundary of $S(\mathcal{K})$. Once they are, the thinning process terminates[6]. The topology is still preserved due to line 10. The additional constraint used at line 16 of Algorithm 2 is very simple and it gives acceptable results in practice. It may be easily coupled with other techniques to preserve shape already described in literature.

Finally, we discuss the situation when one wants to keep the skeleton attached to some pieces of the external boundary B of the mesh. In this case, when testing whether a top dimensional cell T is simple, one should consider $B \cap bd\ T$ as elements in $S(\mathcal{K})$. In other words, elements from B are not considered as an interface between the object to skeletonize and its exterior.

A. Proofs

Now we are ready to give a formal definition of skeleton.

Definition 4.1: A skeleton of a cell complex $\mathcal{K}$, denoted by $S(\mathcal{K})$, is a set of top dimensional cells such that:

1) *$S(\mathcal{K})$ is obtained from $\mathcal{K}$ by iteratively removing top dimensional cells $T_1, \dots, T_n$, provided that the intersection of T_i with $\mathcal{K} \setminus \bigcup_{j=1}^{i-1} T_j$ complement is acyclic. Consequently, homology groups of $S(\mathcal{K})$ and $\mathcal{K}$ are isomorphic;*
2) *There is no top dimensional cell $T \in S(\mathcal{K})$ that has an acyclic intersection with $S(\mathcal{K})$ complement (i.e. the process of removing such cells has been run as long as possible.)*

We want to point out that sometimes, due to some deep phenomena in simple homotopy theory, some skeleton may be redundant. For instance it is possible to have a skeleton of a 3D ball as Bing's house [73] instead being a single top dimensional element. In general it is impossible to avoid this issue due to some algorithmically intractable problems in topology.

In the follwing, we formally show that the skeleton obtained from Algorithm 1 satisfies Def. 4.1. This fact is shown with a sequence of two simple lemmas.

Lemma 4.2: The homology of $\mathcal{K}$ and $S(\mathcal{K})$ are isomorphic.

Proof: The proof of this lemma is a direct consequence of the Mayer–Vietoris sequence [33]. Let $T_1, \dots, T_n$ be the cells removed during the course of the algorithm execution (enumeration is given by the order they were removed by

[6] Note that it also terminates when in the previous step no cells are removed.

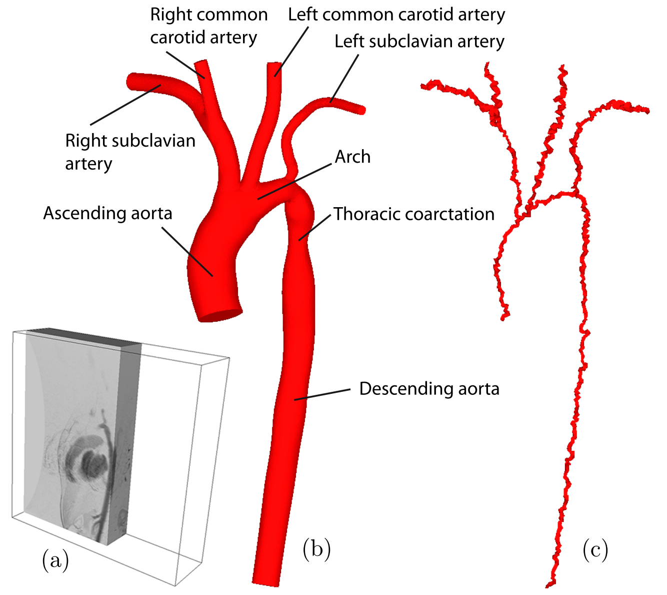

Fig. 4. (a) Magnetic resonance angiography (MRA) image of a moderate thoracic aortic coarctation. (b) Rendering of the 3D triangulated surface (20922 triangles) that represents the patient-specific thoracic aortic coarctation anatomy obtained by segmenting MRA data and (c) the skeleton extracted with Algorithm 2.

the algorithm.) Let us show that, for every $i \in \{1, \ldots, n\}$, homology of $\mathcal{K} \setminus \bigcup_{j=1}^{i-1} T_j$ and homology of $\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j$[7] are isomorphic. Let us write the Mayer–Vietoris sequence in reduced homology for $\mathcal{K} \setminus \bigcup_{j=1}^{i-1} T_j = (\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \cup T_i$:

$$\ldots \to \overline{H}_n((\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \cap T_i) \to$$

$$\to \overline{H}_n(\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \oplus \overline{H}_n(T_i) \to \overline{H}_n(\mathcal{K} \setminus \bigcup_{j=1}^{i-1} T_j) \to \ldots.$$

The intersection $(\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \cap T_i$ is acyclic. This is because the intersection of T_i with the set complement is checked in the acyclicity tables to be acyclic. Once it is, also $(\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \cap T_i$ is acyclic. Therefore, $\overline{H}_n((\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j) \cap T_i)$ is trivial. Also, since T_i is a simplex or cube, it is acyclic. This provides $\overline{H}_n(T_i)$ being trivial (we are considering reduced homology.) Consequently from the exactness of the sequence we have the desired isomorphism between $\overline{H}_n(\mathcal{K} \setminus \bigcup_{j=1}^{i} T_j)$ and $\overline{H}_n(\mathcal{K} \setminus \bigcup_{j=1}^{i-1} T_j)$. The conclusion follows from a simple induction. □

Lemma 4.3: After termination of the algorithm there is no element $T \in S(\mathcal{K})$ that has an acyclic intersection with the $S(\mathcal{K})$ complement.

Proof: Let $T_1, \ldots, T_n$ be the elements removed during the algorithm execution (enumeration is given by the order they were removed.) Suppose, by contrary, that a $T \in S(\mathcal{K})$ exists such that it has an acyclic intersection with the $S(\mathcal{K})$ complement. Let $i \in \{1, \ldots, n\}$ denotes the index of last element among $T_1, \ldots, T_n$ that has nonempty intersection with T. If $i = 0$, then T would be put to the queue L in the line 5 of Algorithm 1 and removed from $S(\mathcal{K})$ in the line 11 of the algorithm, since no change to its intersection with $S(\mathcal{K})$ complements is made by removing $T_1, \ldots, T_n$. If $i > 0$, then after removing T_i the intersection of T with $S(\mathcal{K})$ complement does not change. Therefore, it is acyclic after removing T_j for $j \leq i$. When Algorithm 1 removes T_i in the line 12, T is added to the list L and it is going to be removed in the line 11, since removing T_j for $j > i$ does not affect the acyclicity of the intersection of T with the $S(\mathcal{K})$ complement. In both cases we showed that T is removed from $S(\mathcal{K})$ by Algorithm 1. Therefore, a contradiction is obtained. □

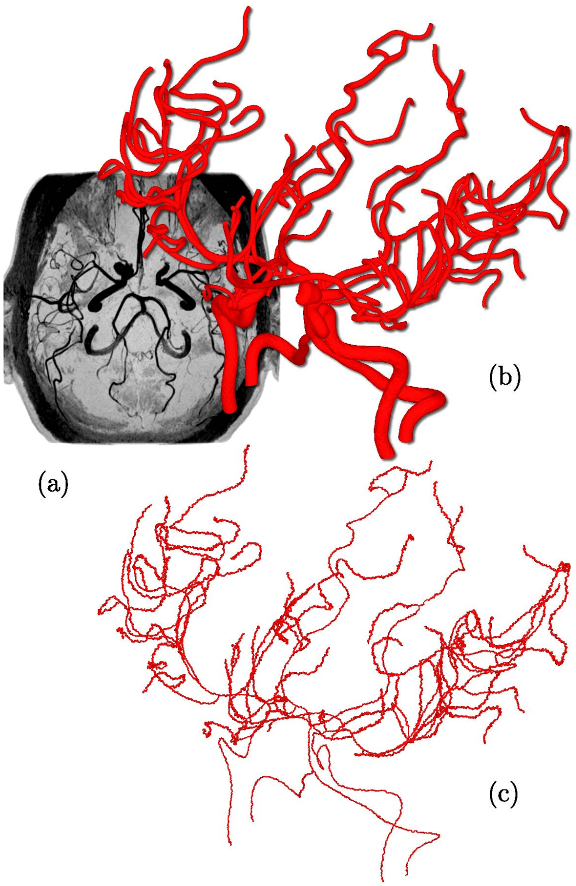

Fig. 5. (a) Magnetic resonance imaging (MRI) image of the cerebral circulation. (b) Rendering of the 3D triangulated surface (196,056 triangles) representing the cerebral circulation obtained by segmenting the MRI data and (c) the skeleton extracted with Algorithm 2.

V. Experimental Results

This section presents some experimental results obtained by running the algorithms proposed in this paper on medical data sets. Algorithm 2 is used when the input cell complex is homologically trivial. In fact, if Algorithm 1 is executed on a trivial complex it outputs in most cases a single top dimensional cell. Algorithm 1 is used when one is interested in the topology preservation only.

A. Assessing Aortic Coarctation and Aneurism

Skeletons can be used in computer-aided diagnostic tools for coarctation and aneurism, by evaluating the transverse areas of any vessel structure, see [23].

[7] The difference used in the formulas in this proof is not a set theoretic difference. All the objects are assumed to contain all their faces.

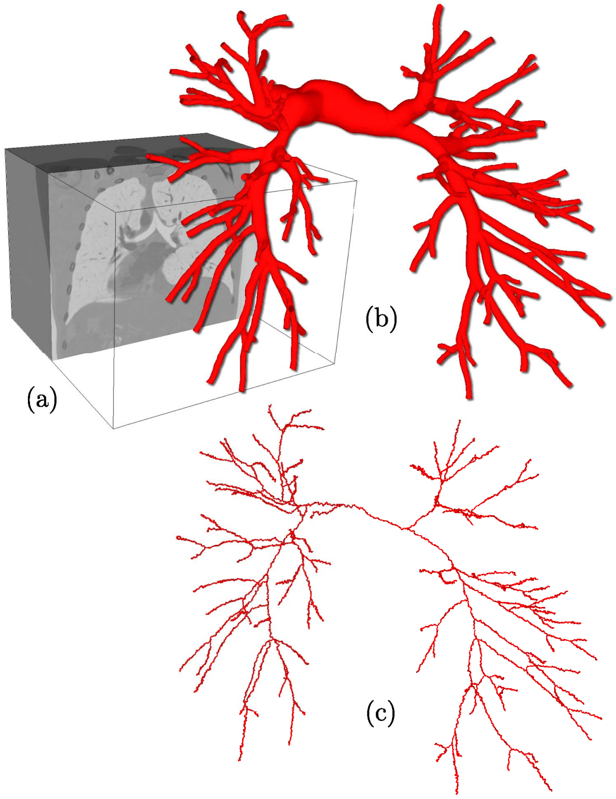

Fig. 6. (a) Computed tomography (CT) image of pulmonary airway trees. (b) Rendering of the 3D triangulated surface (71926 triangles) representing the patient-specific pulmonary arteries and (c) the skeleton extracted with Algorithm 2.

1) Aortic Coarctation: Aorta coarctation is a congenital heart defect consisting of a narrowing of a section of the aorta. Gadolinium-enhanced magnetic resonance angiography (MRA) has been used in a 8 year old female patient to image a moderate thoracic aortic coarctation, see Fig. 4a. Fig. 4b shows a rendering of the 3D triangulated surface, obtained by segmenting the MRA data, which models the ascending aorta, arch, descending aorta, and upper branch vessels. The interior of the surface has been covered with 94756 tetrahedra. The skeleton of this vessel structure, obtained with Algorithm 2, is shown in Fig. 4c.

2) Cerebrovascular Aneurism: Cerebrovascular aneurysms are abnormal dilatations of an artery that supplies blood to the brain. Magnetic resonance imaging (MRI) has been used to image the cerebral circulation in a 47 year old female patient, see Fig. 5a. Fig. 5b shows a rendering of the 3D triangulated surface, obtained from the segmentation of the MRI data. The interior of the surface has been covered with 390,081 tetrahedra. The skeleton of this vessel structure, obtained with Algorithm 2, is shown in Fig. 5c.

B. Analysis of Pulmonary Airway Trees

Skeletons have been used for quantitative analysis of intrathoracic airway trees in [21]. A 3D triangulated surface, shown in Fig. 6b, represents the 3D model of pulmonary airway trees of a 16 year old male patient obtained by segmenting data from computed tomography (CT) images, see Fig. 6a. The interior of the surface is covered with 236,433 tetrahedra. The topology preserving skeleton obtained by Algorithm 2 is shown in Fig. 6c.

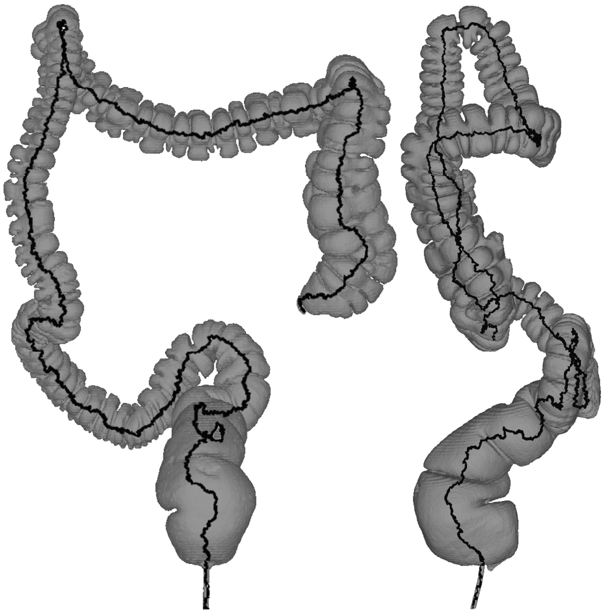

Fig. 7. Rendering of the 3D triangulated surface (506,188 triangles) representing the patient-specific colon anatomy obtained by segmenting CT colonography data and, in black, the skeleton extracted with Algorithm 2.

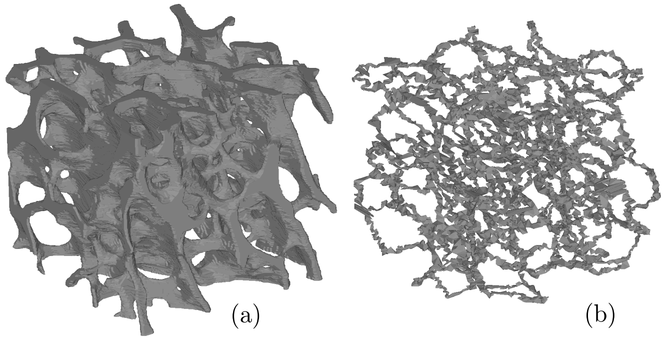

Fig. 8. (a) Rendering of the 3D triangulated surface (285,346 triangles) representing the anatomy of a human bone in a region of interest of the trabecular region obtained by segmenting MicroCT data and (b) the skeleton extracted with Algorithm 1.

C. Extracting Centerline for Virtual Colonoscopy

A 3D triangulated surface that represents a colon is obtained by segmenting data from CT images, see Fig. 7. The interior of the surface is covered with 2,108,424 tetrahedra. The topology preserving skeleton obtained by Algorithm 2, which may be used as a colon centerline to guide a virtual colonoscopy, is shown in black in Fig. 7.

D. Computing Topological Interconnections of Bone Structure

A 3D model of a human bone belonging to a 61 year old male patient has been obtained from a stack of thresholded 2D images acquired by X-ray MicroCT scanning [74]. A stack of 195 2D images has been considered, resulting in a volume of interest of about 4mm×4mm×4mm. From this 3D model, consisting of about 2.4 millions voxels, a 3D triangulated surface has been obtained, see Fig. 8a. The interior of this surface is covered with 688,773 tetrahedra. The topology preserving skeleton obtained by Algorithm 1 is shown in Fig. 8b.

VI. Conclusion

This paper introduces a topology preserving thinning algorithm for cell complexes based on iteratively culling simple cells. Simple cells, that may be seen as a generalization of simple points in digital topology, are characterized with homology theory. Instead of resorting to complicated rule-based approaches, one can detect simple cells with homology computations. The main idea of this paper is to introduce the acyclicity tables which give a classification of all possible simple cells that can occur in a cell complex. These tables are filled in advance automatically by means of homology computations for all possible configurations. Once the acyclicity tables are available, implementing a thinning algorithm does not require any prior knowledge of homology theory or being able to compute homology. The fact that acyclicity tables are filled automatically and correctly for all possible configurations provides a rigorous computer-assisted mathematical proof that the homology-based thinning algorithm preserves topology. We believe that such rigorous topological tools simplify the study of thinning algorithms and provide a clear and safe way of obtaining skeletons.

References

[1] L. Lam, S.-W. Lee, and C. Y. Suen, "Thinning methodologies—A comprehensive survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 869–885, Sep. 1992.

[2] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," in *Proc. Shape Model. Int.*, 2003, pp. 130–139.

[3] N. D. Cornea, D. Silver, and P. Min, "Curve-skeleton properties, applications, and algorithms," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 3, pp. 530–548, May/Jun. 2007.

[4] T. K. Dey, K. Li, J. Sun, and D. Cohen-Steiner, "Computing geometry-aware handle and tunnel loops in 3D models," *ACM Trans. Graph.*, vol. 27, no. 3, p. 45, 2008.

[5] T. K. Dey, K. Li, and J. Sun, "Computing handle and tunnel loops with knot linking," *Comput.-Aided Design*, vol. 41, no. 10, pp. 730–738, 2009.

[6] Q.-Y. Zhou, T. Ju, and S.-M. Hu, "Topology repair of solid models using skeletons," *IEEE Trans. Vis. Comput. Graphics*, vol. 13, no. 4, pp. 675–685, Jul./Aug. 2007.

[7] D. S. Paik, C. F. Beaulieu, R. B. Jeffrey, G. D. Rubin, and S. Napel, "Automated flight path planning for virtual endoscopy," *Med. Phys.*, vol. 25, no. 5, pp. 629–637, 1998.

[8] Y. Ge, D. R. Stelts, J. Wang, and D. J. Vining, "Computing the centerline of a colon: A robust and efficient method based on 3D skeletons," *J. Comput. Assist. Tomogr.*, vol. 23, no. 5, pp. 786–794, 1999.

[9] M. Wan, Z. Liang, Q. Ke, L. Hong, I. Bitter, and A. Kaufman, "Automatic centerline extraction for virtual colonoscopy," *IEEE Trans. Med. Imag.*, vol. 21, no. 12, pp. 14550–1460, Dec. 2002.

[10] R. L. Van Uitert and R. M. Summers, "Automatic correction of level set based subvoxel precise centerlines for virtual colonoscopy using the colon outer wall," *IEEE Trans. Med. Imag.*, vol. 26, no. 8, pp. 1069–1078, Aug. 2007.

[11] M. Ding, R. Tong, S. H. Liao, and J. Dong, "An extension to 3D topological thinning method based on LUT for colon centerline extraction," *Comput. Methods Programs Biomed.*, vol. 94, no. 1, pp. 39–47, 2009.

[12] A. Huang *et al.*, "On concise 3D simple point characterizations: A marching cubes paradigm," *IEEE Trans. Med. Imag.*, vol. 28, no. 1, pp. 43–51, Jan. 2009.

[13] A. P. Kiraly, J. P. Helferty, E. A. Hoffman, G. McLennan, and W. E. Higgins, "Three-dimensional path planning for virtual bronchoscopy," *IEEE Trans. Med. Imag.*, vol. 23, no. 11, pp. 1365–1379, Nov. 2004.

[14] K. Haris, S. N. Efstratiadis, N. Maglaveras, C. Pappas, J. Gourassas, and G. Louridas, "Model-based morphological segmentation and labeling of coronary angiograms," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 1003–1015, Oct. 1999.

[15] A. F. Frangi, W. J. Niessen, R. M. Hoogeveen, T. Van Walsum, and M. A. Viergever, "Model-based quantitation of 3D magnetic resonance angiographic images," *IEEE Trans. Med. Imag.*, vol. 18, no. 10, pp. 946–956, Oct. 1999.

[16] M. Niethammer, A. N. Stein, W. D. Kalies, P. Pilarczyk, K. Mischaikow, and A. Tannenbaum, "Analysis of blood vessel topology by cubical homology," in *Proc. Int. Conf. Image Process.*, vol. 2. 2002, pp. 969–972.

[17] S. R. Aylward and E. Bullitt, "Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction," *IEEE Trans. Med. Imag.*, vol. 21, no. 2, pp. 61–75, Feb. 2002.

[18] S. R. Aylward, J. Jomier, S. Weeks, and E. Bullitt, "Registration and analysis of vascular images," *Int. J. Comput. Vis.*, vol. 55, nos. 2–3, pp. 123–138, 2003.

[19] M. Straka *et al.*, "The vesselglyph: Focus & context visualization in CT-angiography," in *Proc. IEEE Vis.*, Oct. 2004, pp. 385–392.

[20] M. Schaap *et al.*, "Standardized evaluation methodology and reference database for evaluating coronary artery centerline extraction algorithms," *Med. Image Anal.*, vol. 13, no. 5, pp. 701–714, 2009.

[21] J. Tschirren, G. McLennan, K. Palágyi, E. A. Hoffman, and M. Sonka, "Matching and anatomical labeling of human airway tree," *IEEE Trans. Med. Imag.*, vol. 24, no. 12, pp. 1540–1547, Dec. 2005.

[22] M. L. Baker *et al.*, "Modeling protein structure at near atomic resolutions with Gorgon," *J. Struct. Biol.*, vol. 174, no. 2, pp. 360–373, 2011.

[23] K. Kitamura, J. M. Tobis, and J. Sklansky, "Estimating the 3D skeletons and transverse areas of coronary arteries from biplane angiograms," *IEEE Trans. Med. Imag.*, vol. 7, no. 3, pp. 173–187, Sep. 1988.

[24] C. Kirbas and F. Quek, "A review of vessel extraction techniques and algorithms," *ACM Comput. Surv.*, vol. 36, no. 2, pp. 81–121, 2004.

[25] Y. Yang, L. Zhu, S. Haker, A. R. Tannenbaum, and D. P. Giddens, "Harmonic skeleton guided evaluation of stenoses in human coronary arteries," in *Proc. 8th Int. Conf. Med. Image Comput.-Assist. Intervent.*, vol. 8. 2005, pp. 490–497.

[26] E. Sorantin *et al.*, "Spiral-CT-based assessment of tracheal stenoses using 3D-skeletonization," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 263–273, Mar. 2002.

[27] J. T. Ferrucci, "Colon cancer screening with virtual colonoscopy: Promise, polyps, politics," *Amer. J. Roentgenol.*, vol. 177, no. 5, pp. 975–988, 2001.

[28] D. Ravanelli *et al.*, "A novel skeleton based quantification and 3D volumetric visualization of left atrium fibrosis using late gadolinium enhancement magnetic resonance imaging," *IEEE Trans. Med. Imag.*, vol. 33, no. 2, pp. 566–576, Feb. 2014.

[29] B. R. Gomberg, P. K. Saha, H. K. Song, S. N. Hwang, and F. W. Wehrli, "Topological analysis of trabecular bone MR images," *IEEE Trans. Med. Imag.*, vol. 19, no. 3, pp. 166–174, Mar. 2000.

[30] F. W. Wehrli, B. R. Gomberg, P. K. Saha, H. K. Song, S. N. Hwang, and P. J. Snyder, "Digital topological analysis of *in vivo* magnetic resonance microimages of trabecular bone reveals structural implications of osteoporosis," *J. Bone Mineral Res.*, vol. 16, no. 8, pp. 1520–1531, 2001.

[31] P. K. Saha, Y. Xu, H. Duan, A. Heiner, and G. Liang, "Volumetric topological analysis: A novel approach for trabecular bone classification on the continuum between plates and rods," *IEEE Trans. Med. Imag.*, vol. 29, no. 11, pp. 1821–1838, Nov. 2010.

[32] M. Couprie and G. Bertrand, "New characterizations of simple points in 2D, 3D, and 4D discrete spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 637–648, Apr. 2009.

[33] A. Hatcher, *Algebraic Topology*. Cambridge, U.K.: Cambridge Univ. Press, 2002.

[34] M. M. Cohen, *A Course in Simple Homotopy Theory* (Graduate Texts in Mathematics), vol. 10. New York, NY, USA: Springer-Verlag, 1973.

[35] M. Niethammer, W. D. Kalies, K. Mischaikow, and A. Tannenbaum, "On the detection of simple points in higher dimensions using cubical homology," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2462–2469, Aug. 2006.

[36] P. J. Frey and P.-L. George, *Mesh Generation: Application to Finite Elements*. Oxford, U.K.: Hermes Sci. Pub., 2000.

[37] T. Y. Kong and A. Rosenfeld, "Digital topology: Introduction and survey," *Comput. Vis., Graph., Image Process.*, vol. 46, no. 3, pp. 357–393, 1989.

[38] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, 1984.

[39] C. M. Holt, A. Stewart, M. Clint, and R. H. Perrott, "An improved parallel thinning algorithm," *Commun. ACM*, vol. 30, no. 2, pp. 156–160, 1987.

[40] B. K. Jang and R. T. Chin, "Analysis of thinning algorithms using mathematical morphology," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 6, pp. 541–551, Jun. 1990.

[41] M. Ahmed and R. Ward, "A rotation invariant rule-based thinning algorithm for character recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 12, pp. 1672–1678, Dec. 2002.

[42] P. I. Rockett, "An improved rotation-invariant thinning algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1671–1674, Oct. 2005.

[43] M. Ashwin and G. D. Babu, "A new sequential thinning algorithm to preserve topology and geometry of the image," *Int. J. Math. Trends Technol.*, vol. 2, no. 2, pp. 1–5, 2011.

[44] S. Lobregt, P. W. Verbeek, and F. C. A. Groen, "Three-dimensional skeletonization: Principle and algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-2, no. 1, pp. 75–77, Jan. 1980.

[45] T. C. Lee, R. L. Kashyap, and C. N. Chu, "Building skeleton models via 3D medial surface axis thinning algorithms," *CVGIP, Graph. Models Image Process.*, vol. 56, no. 6, pp. 462–478, 1994.

[46] P. P. Jonker and O. Vermeij, "On skeletonization in 4D images," in *Proc. 6th Int. Workshop Adv. Struct. Syntactical Pattern Recognit. (SSPR)*, 1996, pp. 79–89.

[47] C. Pudney, "Distance-ordered homotopic thinning: A skeletonization algorithm for 3D digital images," *Comput. Vis. Image Understand.*, vol. 72, no. 3, pp. 404–413, 1998.

[48] G. Borgefors, I. Nyström, and G. S. Di Baja, "Computing skeletons in three dimensions," *Pattern Recognit.*, vol. 32, no. 7, pp. 1225–1236, 1999.

[49] P. K. Saha, B. B. Chaudhuri, and D. D. Majumder, "A new shape preserving parallel thinning algorithm for 3D digital images," *Pattern Recognit.*, vol. 30, no. 12, pp. 1939–1955, 1997.

[50] K. Palágyi and A. Kuba, "A parallel 3D 12-subiteration thinning algorithm," *Graph. Models Image Process.*, vol. 61, no. 4, pp. 199–221, 1999.

[51] K. Palágyi *et al.*, "A sequential 3D thinning algorithm and its medical applications," in *Information Processing in Medical Imaging* (Lecture Notes in Computer Science), vol. 2082. Berlin, Germany: Springer-Verlag, 2001, pp. 409–415.

[52] K. Palágyi, "A 3-subiteration 3D thinning algorithm for extracting medial surfaces," *Pattern Recognit. Lett.*, vol. 23, no. 6, pp. 663–675, 2002.

[53] W. Xie, R. P. Thompson, and R. Perucchio, "A topology-preserving parallel 3D thinning algorithm for extracting the curve skeleton," *Pattern Recognit.*, vol. 36, no. 7, pp. 1529–1544, 2003.

[54] T. A. Aberra, "Topology preserving skeletonization of 2D and 3D binary images," M.S. thesis, Dept. Math., Tech. Univ. Kaiserlautern, Kaiserslautern, Germany, 2004.

[55] T. Ju, M. L. Baker, and W. Chiu, "Computing a family of skeletons of volumetric models for shape description," *Comput.-Aided Design*, vol. 39, no. 5, pp. 352–360, 2007.

[56] C. Lohou and J. Dehos, "Automatic correction of Ma and Sonka's thinning algorithm using P-simple points," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 1148–1152, Jun. 2010.

[57] G. Németh, P. Kardos, and K. Palágyi, "Thinning combined with iteration-by-iteration smoothing for 3D binary images," *Graph. Models*, vol. 73, no. 6, pp. 335–345, 2011.

[58] P. Wiederhold and S. Morales, "Thinning on quadratic, triangular, and hexagonal cell complexes," in *Combinatorial Image Analysis* (Lecture Notes in Computer Science), vol. 4958. Berlin, Germany: Springer-Verlag, 2008, pp. 13–25.

[59] P. Wiederhold and S. Morales, "Thinning on cell complexes from polygonal tilings, *Discrete Appl. Math.*," vol. 157, no. 16, pp. 3424–3434, 2009.

[60] P. Kardos and K. Palágyi, "On topology preservation for triangular thinning algorithms," in *Combinatorial Image Analysis* (Lecture Notes in Computer Science), vol. 7655. Berlin, Germany: Springer-Verlag, 2012, pp. 128–142.

[61] P. Kardos and K. Palágyi, "Topology-preserving hexagonal thinning," *Int. J. Comput. Math.*, vol. 90, no. 8, pp. 1607–1617, 2013.

[62] T. K. Dey and J. Sun, "Defining and computing curve-skeletons with medial geodesic function," in *Proc. 4th Eurograph. Symp. Geometry Process.*, 2006, pp. 143–152.

[63] L. Tcherniavski and P. Stelldinger, "A thinning algorithm for topologically correct 3D surface reconstruction," in *Proc. 8th IASTED Int. Conf. Vis., Imag., Image Process. (VIIP)*, 2008, p. 119.

[64] J. Cao, A. Tagliasacchi, M. Olson, H. Zhang, and Z. Su, "Point cloud skeletons via laplacian based contraction," in *Proc. Shape Model. Int. Conf. (SMI)*, Jun. 2010, pp. 187–197.

[65] L. Liu, E. W. Chambers, D. Letscher, and T. Ju, "A simple and robust thinning algorithm on cell complexes," *Comput. Graph. Forum*, vol. 29, no. 7, pp. 2253–2260, 2010.

[66] P. Dłotko and R. Specogna. *Thin it Code*. [Online]. Available: http://www.thinit.org, accessed Mar. 15, 2013.

[67] A. A. Markov, "Insolubility of the problem of homeomorphy," *Dokl. Akad. Nauk SSSR*, vol. 121, no. 195, p. 8, 1958.

[68] P. Dłotko and R. Specogna, "Physics inspired algorithms for (co)homology computations of three-dimensional combinatorial manifolds with boundary," *Comput. Phys. Commun.*, vol. 184, no. 10, pp. 2257–2266, 2013.

[69] P. Dłotko and R. Specogna, "Cohomology in 3D magneto-quasistatics modeling," *Commun. Comput. Phys.*, vol. 14, no. 1, pp. 48–76, 2013.

[70] *Computer Assisted Proofs in Dynamic*. [Online]. Available: http://www.redhom.ii.uj.edu.pl, accessed 2005.

[71] M. Mrozek, P. Pilarczyk, and N. Żelazna, "Homology algorithm based on acyclic subspace," *Comput. Math. Appl.*, vol. 55, no. 11, pp. 2395–2412, 2008.

[72] P. Brendel, P. Dłotko, M. Mrozek, and N. Żelazna, "Homology computations via acyclic subspace," in *Computational Topology in Image Context* (Lecture Notes in Computer Science), vol. 7309. Berlin, Germany: Springer-Verlag, 2012, pp. 117–127.

[73] R. H. Bing, *Some Aspects of the Topology of 3-Manifolds Related to the Poincaré Conjecture* (Lectures on Modern Mathematics), vol. 2, T. L. Saaty, Ed. New York, NY, USA: Wiley, pp. 93–128, 1964.

[74] E. Perilli and F. Baruffaldi, "Proposal for shared collections of X-ray micro CT datasets of bone specimens," in *Proc. ICCB*, Zaragoza, Spain, Sep. 2003. [Online]. Available: http://www.biomedtown.org

Paweł Dłotko received the Ph.D. degree in computer science from Jagiellonian University, Kraków, Poland, in 2012, where he was an Assistant Professor from 2011 to 2012. He is currently a Post-Doctoral Researcher with the University of Pennsylvania, Philadelphia, PA, USA. He has authored or co-authored about 20 scientific papers, mainly in the field of computational topology.

Ruben Specogna (S'05–M'08) received the M.Sc. degree in electronic engineering and the Ph.D. degree in electrical engineering from University of Udine, Udine, Italy, in 2003 and 2007, respectively, where he has been an Assistant Professor since 2008. He has authored over 60 papers in ISI scientific journals, in the field of computational electromagnetism, coupled problems computational topology and computer-aided design, nuclear fusion reactors engineering and design, and lab-on-a-chip biosensors.