AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/1

Michał Postolski*, **, ***, Marcin Janaszewski*, **,
Łukasz Jopek*, **, Laurent Babout*

3D Skeletonization of Pulmonary Airway Tree Structures

1. Introduction

Modern medical computer tomography (CT) which uses multidetector spiral scanners can produce three-dimensional volumetric images of very high quality and simply allows to non-invasive look into inside of a human body. This is a very powerful and useful technique being used in a variety of medical applications. 3D volumetric scans of human organs provide an excellent basis for quantification of anatomical structures, for example airway trees. Automatic quantitative description of an airway tree extracted from volumetric CT data set is useful information supporting the non-invasive diagnosis of bronchial tree pathologies, especially chronic obstructive pulmonary disease (COPD) which is common name for pathological changes characterized by airflow limitation due to different combinations of airway diseases and asthma – one of the most widespread disease in the world. Computer analysis of bronchial tree will allow a doctor to obtain precise data on the airway remodeling, which opens new possibilities such as: early identification of pathological changes, precise treatment control, diagnosis of the airway remodeling reason, development of new drugs, etc. The goal of such research is to build a system for automatic measurements of diameter of an airway lumen and thickness of an airway wall. The system should be useful in everyday clinical routine and due to application of modern effective image processing and analysis algorithms it should provide results of high accuracy.

Quantitative analysis of the human airway trees is a challenge in image processing and analysis. Results obtained in previous work on this subject are still not sufficient and they need improvements [11]. Quantitative description of an airway tree consists in application of several steps: segmentation of the tree, skeletonization, decomposition and anatomical labelling, cross section generation and finally quantitative measurements. Each step needs

* Computer Engineering Department, Technical University of Lodz, Poland
** Department of Expert System and Artificial Intelligence, The College of Computer Science in Lodz, Poland
*** Universite Paris-Est, LIGM, A3SI, ESIEE, Noisy le Grand, France

to use different kinds of image processing algorithms. The most works were performed on different segmentation strategies [6, 9, 13], however skeletonization algorithms are still not enough tested and they require attention.

Skeletonization of 3D volumetric images for instance bronchial tree after segmentation, consists in skeleton generation of the 3D object. A skeleton is a minimal representation of the object geometry which preserves its topology. At present many different skeletonization methods which can be categorized to many different groups were developed. However, they produce skeletons with different characteristics and designed for different purposes. Moreover, skeleton has very large impact on the quality of quantitative measurement of a bronchial tree.

In the paper authors tested three skeletonization methods which are based on thinning – the most popular skeletonization approach. However, each algorithm uses different thinning strategy. Two of them based on cubical complex framework which was never used before in such application. The paper presents basic concepts of the algorithms and discussion about their primary features based on acquired results.

In the next chapter basic notions about volumetric images are presented. Sections 3 and 4 consist of skeleton definition and skeletonization algorithm descriptions. In section 5 test results on real CT images and discussion are presented. Summary and conclusions are presented in section 6.

2. Basic notions

Basic concepts of a volumetric image and volumetric image processing were presented in details elsewhere e.g. [7]. In this subsection only the notions necessary to understand the following parts of the paper are introduced.

The elements of 3D image array are called *voxels (points)*. Each voxel is described by a quadruple (x, y, z, v) where (x, y, z) represents 3D location of the voxel and value v indicates its membership. $v = 0$ means that a voxel belongs to a background, $v = 1$ indicating that a voxel belongs to an object (in our case the entire of an airway tree). Background and object points are also called *white* and *black points* respectively. Following definitions in [7] we can distinguish different adjacency relations of voxels. In 3D space two voxels are said to be *26-adjacent* if they are distinct and each coordinate of one differs from the corresponding coordinate of the other by at most 1; two voxels are *18-adjacent* if they are 26-adjacent and differ in at most two of their coordinates; two voxels are *6-adjacent* if they are 26-adjacent and differ in at most one coordinate (see Fig. 1).

A voxel p is an *n-neighbour* of voxel q if p is n-adjacent to q for n = 6, 18, 26. Following the same notations in [7], for a voxel p a voxel q is called an *F-neighbour, E-neighbour*, or *V-neighbour*, of p if it shares a face, an edge or a vertex, respectively, with voxel p.

We say a set *S* of voxels is n-connected if *S* cannot be partitioned into two subsets that are not n-adjacent to each other. If *p* is a voxel in 3D space then *N*(*p*) denotes the set consisting of p and its 26-neighbours.

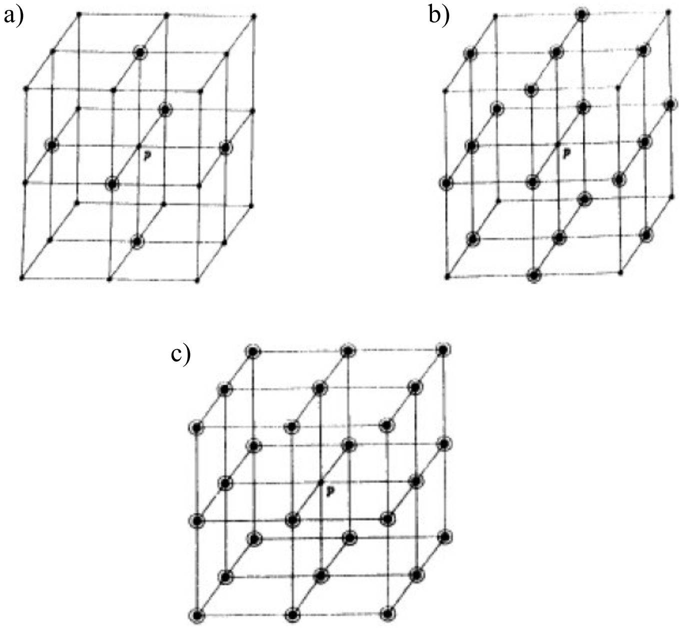

Fig. 1. Different voxels adjacency in 3D images: a) 6-adjacent; b) 18-adjacent; c) 26-adjacent [7]

Taking all above into consideration the 3D dimensional volumetric image *P* can be described as a quadruple $P = (V, m, n, B)$ where $V = Z^3$, $B \subseteq V$ and where $(m, n) = (6, 26), (26, 6), (6, 18), (18, 6)$. We denote by *Z* the set of integers, *V* and *B* denote set of all voxels in the image and a finite set of black voxels respectively if *P* is finite. We say a set *S* of black and/or white points in image *P* is *connected* if *S* cannot be partitioned into two subsets that are not adjacent to each other. A component of a set of black and/or white points *S* is a non-empty connected subset of *S* which is not adjacent to any other point in *S*.

If a black voxel has not a background voxel as a neighbour, it is considered as an *inside voxel* otherwise it is called a *boundary voxel*. A background voxel is called an *outside voxel* if all its neighbours are background voxels. A sequence of voxels $p_1, p_2, ..., p_n$ is called a *voxel path* if it fulfils the following condition: p_i is adjacent to p_j if and only if $|i - j| = 1$, for $i, j = 1, 2, ...,$ n and $i \neq j$. If $p_n = p_1$ then voxel path is *closed path*.

Tunnel in an object *X* is detected whenever there is a closed path in *X* which cannot be transformed into a single point by a sequence of elementary local deformations inside *X* [7]. In 3D space the surrounded (by the object points) connected component of the background is called a *cavity*.

3. Skeleton characteristics

Informally, a skeleton can be defined as a set of curves (centrelines) which passes through the centre of an object interior. Some objects like human colon (see Fig. 2) have only one centreline but there are complicated objects with many branches and holes like human airway trees, blood vessels, cracks in materials and many others where a skeleton can be a very complicated structure.

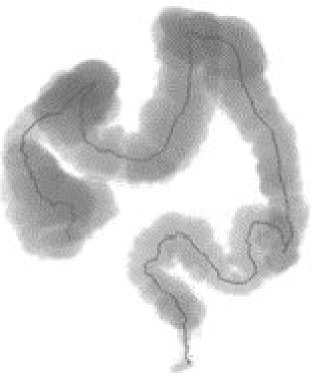

Fig. 2. The human colon with its skeleton superimposed (single centreline)

The notion of a skeleton for any object O was introduced by Blum [3] who defined it as a result of *medial axis transformation*. According to the definition an object voxel p belongs to a skeleton if and only if there is a ball $B(p) \subset O$ centred at a voxel p such that there is not any other ball $B \subset O$ which includes $B(p)$. The skeleton extraction and evaluation based on Blum's definition is very difficult. For instance, it is very difficult to preserve topology and time consuming. Therefore many authors e.g. [10, 15] usually define a skeleton as a set of curves which meets the following conditions:

- *Connectivity* – a skeleton must be a set of connected voxel paths according with the definition presented in the previous section.
- *Centricity* – a skeleton should cross the centre of an object interior. From bronchial tree skeletonization point of view this postulate is very important, because it guarantees that a skeleton is a descriptor of an object centre and can be used to accurately calculate cross sections of an airway tree.
- *Singularity* – a skeleton should be a voxel wide smooth curve without any self-intersections and folds. More formally speaking, a skeleton should be a set of connected voxel paths.
- *Topology preserving* – there are various definitions of topology preserving e.g. [7]. Two objects have the same topology if they have the same number of connected components, holes, and cavities. The model airway tree dataset should consist of one object without any holes and cavities. Therefore the skeleton generation algorithm should extract skeleton without any loops or disconnected segments.

– *Parameterisation* – along with a set of skeleton voxels an algorithm should result with a skeleton structure description or allow to simply generate such description. It means that the main centreline and its branches should be parameterised by their starting voxel, ending voxel and length. Such a parameterisation simplifies further processing like pruning unnecessary branches and allows anatomical labelling of the tree.
– *Robustness* – the algorithm should not be sensitive to little changes in an object structure or geometric transformations such as translation or rotation. What is more extracted skeleton should not fluctuate according to changes of starting and ending voxel.
– *Automation* – The algorithm should extract a skeleton fully automatically without user interaction. It concerns especially automatic determination each of the tree segment starting and ending voxel.
– *Cost effectiveness* – For large and complicated data computational time and memory utilisation are critical. Therefore algorithms should be fast enough to extract a skeleton for complicated airway tree structure data in seconds on standard PC computer.

4. 3D skeletonization algorithms

In the last several decades many skeleton extraction algorithms have been constructed. Most of them generate skeletons for general purposes which can be used in wide range of different applications. However, some of them are dedicated to special task such as centreline generators derived from skeletonization algorithms which were refined to perform well centreline descriptor for volumetric colon images for a virtual colonoscopy application. The main reason to generate algorithms only for special approaches is that universal skeletonization algorithms are not sufficient in all cases or they do not always satisify all conditions (see section 3). This is also the reason why so many skeletonization methods have been constructed so far. These algorithms can be divided into the following groups:

Manual extraction – These methods require significant manual work of a user who is responsible for marking a centre of an object on each image slice of several hundred in a dataset. Then the skeleton is linearly interpolated between consecutive marked points. Unfortunately the method is time consuming, sometimes difficult to perform and does not guarantee the centricity of marked points because of possible human mistakes.

Topological thinning – these algorithms delete, on each iteration, so called simple points from the boundary of an object. A simple point is defined as an object point which deletion does not change the object topology. The process stops when no more simple points to delete is found.

Voxel coding based – A voxel coding scheme is a voxel by voxel recursive propagation and assignment of integer codes to object voxels starting from a set of voxels which are called seeds. Most of these algorithms use a special voxel coding called the distance transform or an approximation of distance transform were the seed set consists of object boundary voxels. Such a distance transform results in an image called a distance field which has very useful property from skeleton generation point of view. Its ridges correspond to the voxels that are local centres in the object. Based on the ridges various algorithms use various approaches to build the skeleton. Usually the set of ridges is pruned and then remaining voxels are connected in order to form one voxel wide connected skeleton. Methods based on potential field are also assigned to this group. The methods take advantage of similar strategy to generate skeleton, however they use different kinds of vector field than distance transform coding.

Hybrid methods – These methods consist of different algorithms which cannot be simply categorized into previous groups. They work in different frameworks than voxel framework for example in cubical complex framework, but use standard thinning scheme. It is also possible, in one algorithm, to use both thinning and distance transform technique. For instance, thinning constrained by medial axis.

In this article authors focus on different thinning approaches and hybrid methods based on thinning which are the most popular group of skeletonization algorithms in real application. In the next subsection we present only general view of algorithms way of working.

4.1. A Fully Parallel Thinning Algorithm

The first presented algorithm, developed by Ma and Sonka [8], belongs to the topological thinning group. We denote this algorithm as FPT. The algorithm tests all border voxels on each iteration. Once a voxel is visited the algorithm checks if it meets at least one of a priori defined deleting constraints. If so the voxel is deleted. The process ends when it does not delete any voxel in the last iteration. Points which are not deleted during the process form the final skeleton. Ma and Sonka's algorithm is based on the fully parallel strategy and uses a set of predefined deleting templates to test neighbourhood of each border voxel. When a voxel and its neighbourhood match at least one template then the voxel is marked to be deleted. After examination of all border voxels the marked ones are deleted by changing their values to 0.

Deleting templates are represented as cubic grids with three types of points (see Fig. 3). An object point and a background point are denoted with "•", and "o" respectively. A "don't care point", which means that it can be either object point or background point, is unmarked. Ma and Sonka presented four classes of deleting templates (A, B, C and D). Figure 4 shows the four basic template cores. The translation of the cores results in deleting templates: six in class A, twelve in class B, eight in class C and twelve in class D [14]. We tested upgraded version of Ma and Sonka's algorithm [14]. It was proved that the original algorithm do not preserve connectivity in specific cases [14]. In new version of the algorithm 12 templates of class D were changed into new 32 ones. This change leads to the connectivity preserving algorithm. What is more in order to preserve topology the algorithm cannot delete so-called tail-points which are defined as line-end points or near-line-end points [14].

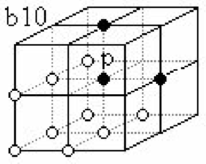

Fig. 3. One of deleting templates [14]

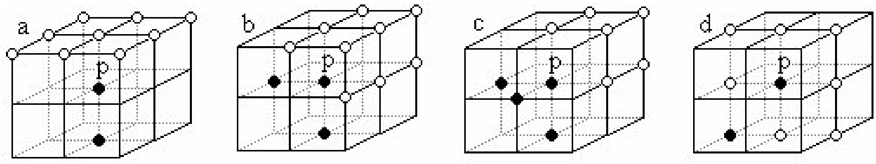

Fig. 4. Four template cores. Class A, B , C and D [14]

Taking into consideration all above the FPT algorithm can be expressed as follows:

Repeat

  Mark every border point of an object

  Repeat

    Simultaneously delete every non tail-point which satisfies at least one deleting template from class A, B, C, or D;

  Until no point can be deleted;

  Release all marked but not deleted points;

Until no marked point can be deleted;

4.2. Curvilinear Thinning Algorithm based on Critical Kernels

The skeleton method presented in this sub-section is denoted CTA hereafter. Some new basic notions have to be introduced to explain its skeletonization strategy. A more extensive description is provided in [1].

We denote by Z the set of integers, N_+ set of positive integers. Let $E = Z^3$. Informally, a *simple point p* of a discrete object $X \subset E$ is a point which is "inessential" to the topology of X. In other words, we can remove the point p from X without "changing the topology of X". Skipping some technical details, let $A(x, X)$ be the set of points of $X \backslash \{x\}$ lying in a neighborhood of x, and let $Ab(x, X)$ be the set of points of the complementary of X (background) lying in a neighborhood of x. Then, $T(x, X)$ (resp. $Tb(x, X)$) is called topological number and denotes the number of connected components of $A(x, X)$ (resp. $Ab(x, X)$). A point $x \in X$ is simple for X if and only if $T(x, X) = Tb(x, X) = 1$. Also, if a point $x \in X$ is such that $Tb(x, X) = 1$, then removing x from X does not create a new tunnel. Let X be any finite subset of E. The subset Y of E is a *homotopic thinning* of X if $Y = X$ or if Y may be obtained from X by iterative deletion of simple points.

The main way of working of CTA algorithm is to delete simple points in parallel until stability. Removing simple points is traditional topological thinning scheme like in FPT algorithm. However, in general, removing at the same time two simple voxels does not guarantee topology preservation. Bertrand and Couprie developed a new framework, the critical kernel framework [2], relying on the cubical complexes, presented shortly in the next subsection, to give a method on how to remove, in the voxel framework, multiple simple points in the same time. Bertrand and Couprie originally proposed CTA and some others generic thinning schemes which allow to compute a wide variety of skeletons based on critical kernels in [2]. Interested reader can find basic notions about critical kernels in [2].

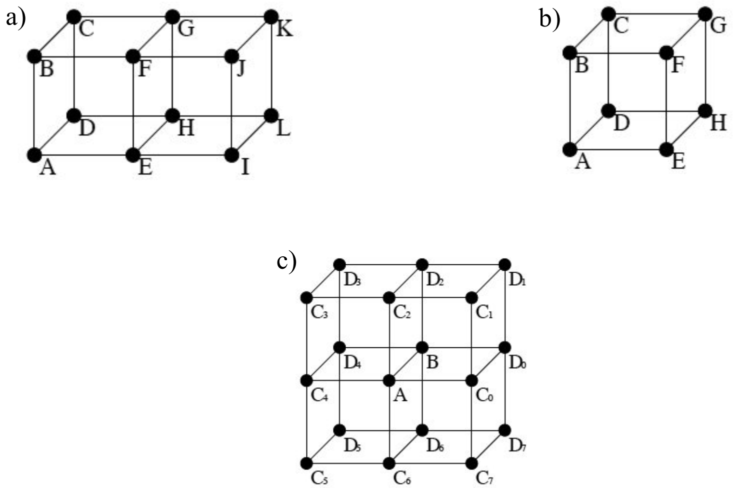

Fig. 5. Set of mask proposed in [2]. Here, voxels are represented by points. Voxel match the mask if it is simple and its neighbourhood match proper configuration of points denoted as letters from A to L. Detailed explanation of different possible points configuration and its consequences are presented in [2]

General way of working of CTA algorithm can be described as follows. First, in each iteration all simple voxels are addressed. Instead of using masks like in the previous approach each voxel is tested by analysing its local topological numbers. The next step should consist in removing all simple voxels in parallel. However to preserve topology additional step is required. In this step thinning is constrained by set of voxels which are essential to preserve topology. This set can be computed by using three mask proposed in [2] (see Fig. 5). Such set of mask were designed thanks to critical kernels framework. For computing curvilinear skeleton all points which are curve points ($T(x, X) = 2$ and $Tb(x, X) = 1$) need to be preserved, therefore they are add as constrained points. At the end all simple voxels which do not belong to the constrain set are deleted.

4.3. Thinning Algorithm based on Cubical Complex

The next algorithm, denoted as CCT hereafter and proposed in [4], is based on cubical complex framework which is an alternative to the voxel framework used in both thinning approaches presented earlier. We only introduce basic definitions of this framework below.

In the three dimensional cubical complex framework, objects are built with various kinds of basic elements: cubes, squares, lines and points. Let Z be the set of integers, we consider the family of sets F_0^1 and F_1^1, such that $F_0^1 = \{\{a\} \mid a \in Z$ and $F_1^1 = \{\{a, a+1\} \mid a \in Z\}$. Any subset *f* of Z^n such that *f* is the Cartesian product of *m* elements of F_1^1 and $(n - m)$ elements of F_0^1 is called a face or an *m*-face of Z^n, *m* is the dimension of *f*, we write $dim(f) = m$. We denote by F^n the set composed of all *m*-faces in Z^n, $m \in \{0, ..., n\}$. A 0-face is called a *vertex*, a 1-face is an *edge*, a 2-face is a *square*, and a 3-face is a *cube* (see Fig. 6).

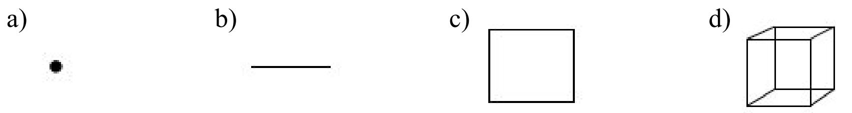

Fig. 6. Graphical representation of: a) a 0-face; b) a 1-face; c) a 2-face; d) a 3-face [2]

Let $f \in F^n$. We set $\hat{f} = \{g \in F^n \mid g \subseteq f\}$, and $\hat{f}^* = \hat{f} \setminus \{f\}$. Any element $\hat{f}$ is a *face* of *f*, and any element of $\hat{f}^*$ is a *proper* face of *f*. We call *star* of *f* the set $\breve{f} = \{g \in F^n \mid f \subseteq g\}$, and we write $\breve{f}^* = \breve{f} \setminus \{f\}$, any element of $\breve{f}$ is a *coface* of *f*. It is plain that $g \in \hat{f}$ if $f \in \breve{g}$.

A set *X* of faces in F^n is a *cell*, or *m-cell*, if there exist an *m*-face $f \in X$ such that $X = \hat{f}$. The *closure* of a set of faces *X* is the set $X^- = \cup\{\hat{f} \mid f \in X\}$. The set $\bar{X}$ is $F^n \setminus X$.

A finite set *X* of faces in F^n is a cubical complex if $X = X^-$, and we write $X \prec F^n$. Any subset *Y* of *X* which is also a complex is a subcomplex of *X*, and we write $Y \prec X$.

Thinning in cubical complex is performed by using elementary operation collapse [4]. It consists in removing two distinct elements (*f*, *g*) from complex *X* under the condition that *g* is contained in *f* and is not contained in any other element of *X*.

Let $X \prec F^n$, and let *f*, *g* be two faces of *X*. The face *g* is free for *X*, and the pair (*f*, *g*) is a free pair for *X* if *f* is the only face of *X* such that *g* is a proper face of *f*.

Let $X \prec F^n$, and let (*f*, *g*) be a free pair for *X*. The complex $X \setminus \{f, g\}$ is an elementary collapse of *X*.

Collapse operation is repeated several times to compute a skeleton. In each iteration parallel removal of simple pairs is performed by rules presented in [4]. The algorithm results in curvilinear skeleton.

5. Test results and discussion

This section presents test results of the above described algorithms. Moreover it contains the discussion of the algorithms properties concluded from the results. All tests have been performed, with the use of ten real human computer tomography chest dataset acquired using GE LightSpeed VCT multidetector CT scanner. The set of stack images is of size 512×512 and voxel dimensions are: $x = y = 0.527$ mm, $z = 0.625$ mm. All of datasets where segmented to extract airway tree structure by using very reliable approach based on hole closing [12]. It is worth emphasizing that application of a reliable segmentation method, prior to skeletonization, was very important, because simple algorithms such as popular standard region growing approach usually generates very distorted airway tree with leaks or with many cavities or holes (see Fig. 7). This makes impossible the generation of proper skeleton and in consequence bias the comparition between skeletonization algorithms.

Algorithm FPT was implemented by authors. Algorithms CTA and CCT where implemented thanks to theirs authors M. Couprie, G. Bertrand and J. Chaussard from University Paris-Est, France. Moreover, CTA algorithm was taken from the image processing library called PINK [16].

All performed tests consist of application of all implemented algorithms to skeletonize all datasets and after that generated skeleton properties where checked. Since quantitative descriptors of skeleton properties is very hard to achive all of generated skeletons were only visually checked to apprehend if they meet all conditions of good skeleton principles described in section 3. Centricity feature were analysed by superimposing skeleton and pruned medial axis of an object. The medial axis was pruned to preserve only thin segments exactly in the centre of an input object [5]. Such a strategy makes the pruned medial axis a good descriptor of an object centre.

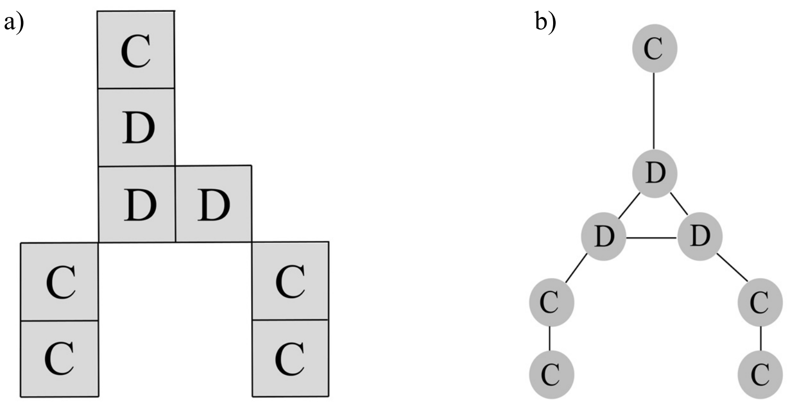

Fig. 7. A part of skeleton of corresponding airway tree structure (a) structure from a) converted to a graph (b). The generated graph is not equivalent to the real tree structure

The results are presented in Figures 8 and 9. FPT algorithm produces smooth skeleton which crosses the centre of an airway tree. However; FPT algorithm does not generate minimal possible skeleton in all cases, because it also generates some flat, one voxel wide surfaces (see Fig. 9b). A flat surface is the result of inadequate set of deleting templates. If an object is thin and flat or in previous iteration the algorithm generated one voxel thin surfaces then proposed deleting templates can not work properly. In such cases the remaining object voxels do not meet any deleting condition. Therefore the fully parallel algorithm needs significant refinement to meet singularity condition.

CTA algorithm generates minimal possible skeleton which is a very good descriptor of whole tree. Moreover the algorithm reveals little sensitivity to noise and generates all important branches. However, detailed analysis of produced skeleton showed that in some cases the skeleton is not as well centred as the skeleton generated with FPT (see Fig. 9d). In such cases CTA produces a skeleton which does not cover medial axis points, instead the skeleton “oscillates” around such points. It can be important source of measurement errors in further quantitative airway analysis.

a) b) c)

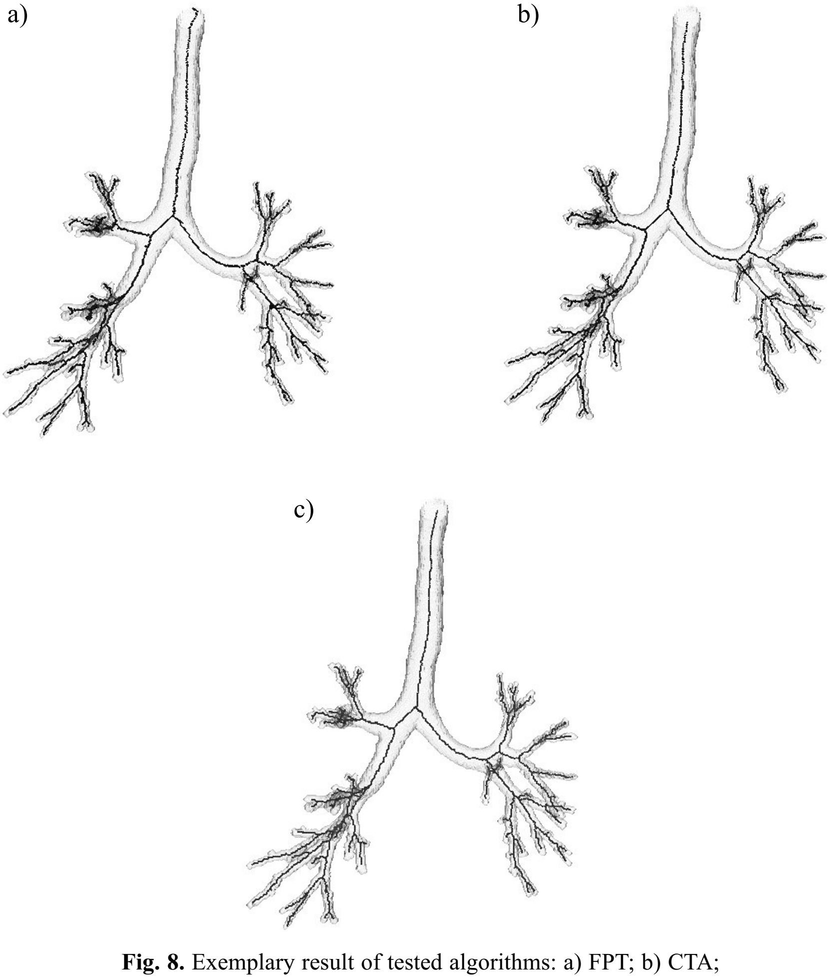

Fig. 8. Exemplary result of tested algorithms: a) FPT; b) CTA;
c) CCT. Generated skeleton is superimposed on the corresponding tree

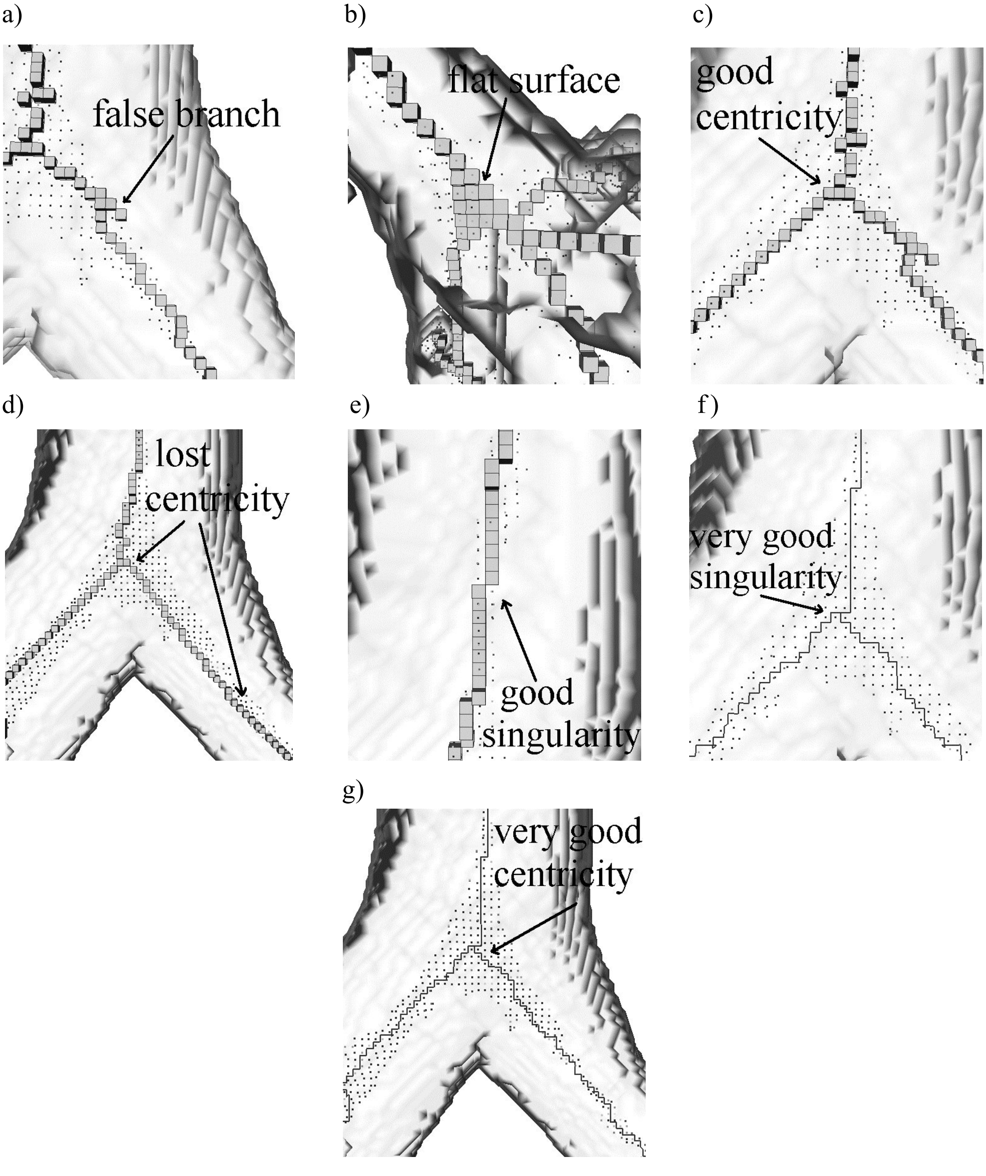

Fig. 9. Visual inspection of interesting parts of generated skeletons.
a–c) FPT; d–e) CTA; f–g) CCT algorithms respectively.
Gray dots denote voxels of pruned medial axis

The last tested algorithm (CCT), uses cubical complex framework and in consequence it generates a skeleton which consists only of edges of voxels. Therefore a skeleton is build using lines instead of cubes (see Fig. 9f). This feature has many interesting advantages from quantitative airway analysis point of view. First of all, generated skeleton has very good centricity feature – is situated exactly at the centre of medial axis points. Similarly to CTA algorithm, CCT method addresses all important branches of a tree and it is not sensitive to the small noise in a tree structure. Moreover a skeleton generated with CCT consists of only lines (no surfaces) so it can be easily converted to a graph which is a very important advantage from bronchial tube analysis point of view, where the graph representation is needed for further analysis. Skeletons generated with the use of voxel framework algorithms need additional post-processing to convert voxel representation into graph representation which is not a simple task. In case of some voxel framework algorithms it is even impossible to create graph representation without loosing centricity or exact representation of an airway tree (see Fig. 7). It means that all tested methods which produce voxels, poorly meet parameterization condition. To conclude, the CCT algorithm meet both parameterisation and automation conditions.

The summary of characteristic of all tested algorithms is presented in table 1. Taking all above into consideration, from author's point of view, the best strategy to skeletonize airway trees is to use novel CCT algorithm which is based on cubical complex framework. It is worth emphasising that, as far as we know, this is the first time where abstract framework based methods are applied to skeletonize bronchial trees.

The computing time of each algorithms were also evaluated. All tested algorithms have a linear time complexity. On standard PC platform computer with Pentium dual core 2GHz processors, each computing time did not exceed several seconds. Unfortunately, exact time comparison between algorithms is useless because of different implementation and strong code optimisation technique in CTA and CCT algorithms. However, computing time is not crucial in quantitative analysis of the human airway tree applications since no characterisation are made in real time.

Table 1
Comparison of skeleton characteristics generated by FPT, CTA, CCT algorithms

	FPT	CTA	CCT
Connectivity	Poor	Very good	Very good
Centricity	Good	Good	Very good
Singularity	Poor	Good	Very good
Topology preserving	Poor	Very good	Very good
Robustness	Poor	Very good	Very good
Parameterisation	Poor	Good	Very good
Automation	Very good	Very good	Very good

6. Conclusion

In the article three skeletonization algorithms which belong to the most popular group (topological thinning) were tested. Algorithms have been used to skeletonize human airway trees segmented from CT images as a part of medical quantitative airway tree analysis. Results showed that algorithms have different characteristic and produce different skeletons. Algorithms FPT and CTA have some drawbacks, especially problems concerning of producing "minimal" possible curvilinear skeleton, achieving very good centricity and they do not meet parameterization features. In consequence they can be source of errors in further quantitative analysis or in some cases make it even impossible. That drawbacks need to be eliminated, to make the algorithms useful in a bronchial tree analysis system, by using sophisticated refinement procedures e.g. [11].

The CCT algorithm is the best and the most interesting one because it has a set of intrinsic properties which are very hard to achieve in voxel framework. Detailed analysis of features of generated skeletons showed that CCT algorithm meet all condition of good skeleton. Moreover, it allows simple decomposition of a tree and its conversion to a graph without risk of changing tree structure. In consequence it can be successfully used in quantitative analysis task and it is very good alternative to traditional voxel framework methods. Moreover, it is the first time that an algorithm based on cubical complex is used in such medical application. It makes cubical complex based methods interesting field of further research.

From authors' point of view, selecting proper skeletonization algorithm is a very important step in building a system for quantitative analysis of airway trees, because accuracy of measurements strongly depends on quality of a generated skeleton. Furthermore, a skeleton is also used for different purposes such as anatomical labeling of a tree or registration and matching two different airways acquired in different time for monitoring progress of pathological changes. Therefore, it is very important to test different skeletonization techniques in real world applications.

In the future the authors plan to compare more skeletonization methods and made some improvements if necessary to obtain skeletons which meet all postulates presented in this paper.

Acknowledgements

The authors would like to thank PINK library developer team and would also thank J. Chaussard from University Paris-Est, France, for his implementation of CCT algorithm.

The authors would like to thank prof. Ludomir Stefańczyk and Mariusz Jędrzejczyk, Medical University of Łódź for providing CT scans to tests and for discussions which have significant influence on this work.

The authors M. Postolski and Ł. Jopek are scholarship holders of project entitled "Innovative education..." supported by European Social Fund.

References

[1] Bertrand G., *Simple points, topological numbers and geodesic neighborhoods in cubic grids.* Pattern Recognition Letters, vol. 15(10), 1994, 1003–1011.

[2] Bertrand G., Couprie M., *A new 3D parallel thinning scheme based on critical kernels.* Lecture Notes in Computer Science, vol. 4245, 2006, 580–591.

[3] Blum H., *Transformation for extraction new descriptions of shape.* Models for the Perceptions of Speech and Visual Form, 1967, 362–380.

[4] Chaussard J., Couprie M., *Surface thinning in 3D cubical complexes.* Lecture Notes in Computer Science, vol. 5852, 2009, 135–148.

[5] Couprie M., Coeurjolly D., Zrour R., *Discrete bisector function and Euclidean skeleton in 2D and 3D.* Image and Vision Computing, vol. 25(10), 2007, 1543–1556.

[6] Graham M.W., Gibbs J.D., Higgins W.E., *Robust system for human airway-tree segmentation.* Proc. SPIE, vol. 6914, 2008.

[7] Kong T.Y., Rosenfeld A., *Digital topology: introduction and survey.* Computer Vision, Graphics, and Image Processing, vol. 48, 1989, 357–393.

[8] Ma C.M., Sonka M., *A fully paralel 3D thinning algorithm and its applications.* Computer Vision, and Image Understanding, vol. 64, 1996, 420–433.

[9] Mayer D., Bartz D., Ley S., Thust S., Heussel C.P., Kauczor HU., Straser W., *Segmentation and virtual exploration of tracheobronchial trees.* International Congress Series, vol. 1256, 2003, 35–40.

[10] Palagyi K., *A 3-subiteration 3D thinning algorithm for extracting medial surfaces.* Pattern Recognition Letters, vol. 23, 2002, 663–675.

[11] Palagyi K., Tschirren J., Hoffman E.A., Sonka M., *Quantitative analysis of pulmonary airway tree structure.* Computers in Biology and Medicine, vol. 36(9), 2006, 974–996.

[12] Postolski M., Janaszewski M., Fabijanska A., Babout L., Couprie M., Jedrzejczyk M., Stefanczyk L., *Reliable airway tree segmentation based on hole closing in bronchial walls.* Advances in Soft Computing: Computer Recognition Systems 3, vol. 57, 2009, 389–396.

[13] Tschirren J., Hoffman E.A., McLennan G., Sonka M., *Intrathoracic airway trees segmentation and airway morphology.* IEEE Trans. Med. Imaging, vol. 24(12), 2005, 1529–1539.

[14] Wang T., Basu A., *A note on a fully parallel 3D thinning algorithm and its applications.* Pattern Recognition Letters, vol. 28(4), 2007, 501–506.

[15] Zhou Y., Toga W., *Efficient skeletonization of volumetric objects.* IEEE Transaction on Visualization and Computer Graphics, vol. 5, 1999, 196–209.

[16] PINK Library http://www.esiee.fr//coupriem.