Comparison of Several Centreline Extraction Algorithms for Virtual Colonoscopy

Marcin Janaszewski[1,2], Michał Postolski[1], Laurent Babout[2], Edward Kącki[1]

[1]Department of Expert Systems and Artificial Intelligence, The College of Computer Science in Lodz, Poland
[2]Department of Computer Engineering, Technical University of Lodz, Poland
janasz@kis.p.lodz.pl; postol@wsinf.edu.pl; lbabout@kis.p.lodz.pl; e25kacki@cosmosnet.pl

Abstract. In the paper, authors report on test of three skeletonization algorithms, which could be used as centreline generators for 3D colon images. Two of them belong to the topological thinning group of skeletonization algorithms and the last one to the distance mapping group. After adaptation to centreline generation task the algorithms were tested on a real 3D colon image and obtained results are reported along with the characteristics of each algorithm performance. What is more the authors have made some improvements to the algorithms in order to obtain better results. The improved algorithms were also tested and results are reported. Moreover the paper contains comparison of the new algorithms with their original counterparts. Final discussion and presentation of future works are also included in the paper.

Keywords: Skeletonization, virtual colonoscopy, thinning algorithm, subiteration algorithm, voxel coding, centreline extraction.

1 Introduction

Visual endoscopy uses medical imaging, computer graphics and image processing technologies to examine the interior structures of human organs [3, 6]. The technique is superior to the traditional fiberoptic endoscopy because of its non-invasivity, cost-effectiveness and high accuracy. Moreover it is free of risk and side effects like perforation or infection and can be applied for some special organs which are impossible to inspect using a traditional endoscopy (e.g. blood vessels). Therefore, many prototype systems have been developed for a variety of clinical applications, including virtual colonoscopy (VC) [7], virtual bronchoscopy [13], virtual angioscopy [2].

VC procedure is quite complicated. Firstly the system takes computed tomography scan of a patient's abdomen. As a result doctors obtain several hundred high-resolution slice images, which are quickly taken during a single breath hold, forming a volumetric abdomen dataset. Then, using image segmentation algorithms, the entire colon is extracted from the abdomen dataset. After that, the colon centreline is extracted using 3D image processing algorithms. Intuitionally, centreline is a curve

which goes through the centre of an organ. Then the system is ready to perform real time navigation based on volume rendering on a personal computer. The inspection can be realized in an automatic manner, following the centreline or by interactive navigation for more accurate study of suspicious areas. A crucial component of a VC system is the extraction of the centreline. It plays several roles from providing compact colon shape description and accurate colon geometry measurement to supporting path generation for both interactive and automatic navigations. The centreline generation has been a challenging problem due to the colon complexity, presence of distortion in 3D colon images and several postulates which a centreline should meet. Therefore many centreline extraction algorithms have been developed for the last 2 decades. Most of them are derived from skeletonization algorithms because a centreline can be treated as a part of a skeleton. Unfortunately, despite of several decades of research in this field, there is not a centreline generation algorithm which is superior to all others and generates centrelines which meet all desired postulates. The literature analysis leads to the conclusion that despite of the fact that many useful algorithms have been constructed there is still the possibility to build new, faster and more accurate centreline generators e.g. algorithms which incorporate the best properties of several existing solutions. What is more, there are some skeletonization algorithms which have not been extensively tested on 3D colon images. Therefore we attempt to test three skeletonization algorithms, presented in [10, 12, 14] respectively, which application to centreline generation for VC has not been widely reported and discussed in the literature.

2 Basic Concepts of 3D Volumetric Images

Basic concepts of a volumetric image and volumetric image processing were presented in detail elsewhere e.g. [14]. In this subsection only the notions necessary to understand the following parts of the paper are introduced.

The 3D crack images are 3D binary volume data sets. A 3D binary volume set consists of voxels – the smallest unit cube in the volume. 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 a colon). Treating a voxel as a unit cube results in three kinds of voxel neighbourhood. Following the same notations in [14], for a voxel p a voxel q is called a *F-neighbour, E-neighbour*, or *V-neighbour*, of p if it shares a face, an edge or a vertex, respectively, with voxel p. Voxels p and q are also called *F-connected*, *E-connected*, *V-connected*. Two voxels are *adjacent* or *neighbours* if they are at least V-connected. If a voxel within an object has 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; otherwise it is regarded as a boundary voxel. 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 $i \neq j$.

A set of voxels is *connected* if, for any two voxels within it, there is a path within the set connecting them. Two sets of voxels A, B are *connected* or *adjacent* if there

are at least two voxels, one within set A and the other one within set B, which are neighbours.

3 Centreline Characteristics

Informally, a centreline can be defined as a curve which passes through the centre of an object interior. Some objects like colon have only one centreline but there are complicated objects with many branches like blood vessels, human lungs, cracks in materials and many others.

The notion of a centreline for any object O was introduced by Blum [1] who defined it as a result of *medial axis transformation*. According to the definition an object voxel p belongs to a centreline 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 centreline extraction and evaluation based on Blum's definition is very difficult and time consuming therefore many authors e.g. [3, 6, 8, 14] usually define a centreline as a curve which meets the following conditions:

1. *Connectivity* – a centreline must be a voxel path according with the definition presented in the previous chapter.
2. *Centricity* – a centreline should traverse a centre of an object interior. The postulate guarantee that a centreline is not only a descriptor of an object centre but also is a safe path for colon navigation preventing a navigator from hugging a colon walls on sharp bends.
3. *Singularity* – centreline should be one wide smooth curve without any self-intersections and folds. More formal a centreline should be a voxel path.
4. *Topology preserving* – there are various definitions of topology preserving e.g. [4, 11]. Two objects have the same topology if they have the same number of connected components, holes, and cavities. The model colon dataset should consist of one object without any holes and branches. Nevertheless, in real situations narrow bowel can collapse or by twisting various colon parts can touch each other resulting in holes, branches or even separation a colon into several objects. Therefore the centreline generation algorithm should detect disconnected colon segments, branches and holes. In case of several segments the algorithm should extract centreline for each segment. In case of a hole the classical skeletonization algorithms generate a loop which consists of two connected branches, but one of the branches is false because it traverses the two walls touching area. Therefore an algorithm should detect and discard such a branch.
5. *Parameterisation* – along with a set of centreline voxels an algorithm should result with a centreline structure description. It means that the main centreline and its branches should be parameterised by their starting voxel, ending voxel and length. Such a parameterisation simplify further processing like refinement, hole detection or pruning unnecessary branches.
6. *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 centreline should not fluctuate according to changes of starting and ending voxel.

7. *Automation* – The algorithm should extract a centreline fully automatically without user interaction. It concerns especially automatic determination of a colon starting and ending voxel.
8. *Cost effectiveness* – For large and complicated data computational time and memory utilisation are critical. Therefore algorithms should be fast enough to extract a centreline for complicated colon data in seconds on standard PC computer.

4 Characteristics of Virtual Colonoscopy Algorithms

In the last several decades many centreline extraction algorithms dedicated to VC have been constructed. Most of them derive from skeletonization algorithms which were refined to perform well for volumetric colon images. These algorithms can be divided into three groups:

1. *Manual extraction* – These methods require significant manual work of user who is responsible for marking a centre of a colon on each image slice of several hundreds in a dataset. Then the centreline 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.
2. *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 deletion.
3. *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 centreline 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 centreline.

4.1 A Fully Parallel Thinning Algorithm

The first tested algorithm, developed by Ma and Sonka [5], belongs to the topological thinning group. In this section we present only general view of its way of working. 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 which 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 Figure 1). 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 2 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 [12]. We tested upgraded version of Ma and Sonka's algorithm [12]. It was proved that the original algorithm do not preserve connectivity in specific cases [12]. 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 [12].

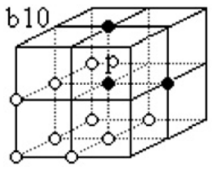

Fig. 1. One of deleting templates [12].

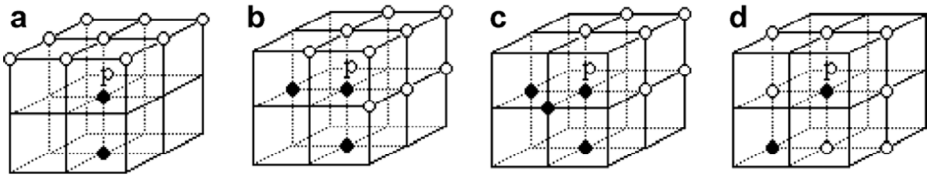

Fig. 2. Four template cores. Class A, B , C and D [12].

Taking into consideration all above the 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 12-Subiteration Thinning Algorithm

The next algorithm which we have tested follows different thinning strategy than the first one. Detailed presentation of the strategy has been published in [9] by Palagyi. In this type of thinning strategy each iteration of a thinning process is divided into subiterations. Common subiteration algorithms use three or six subiterations, however Palagyi proposed an approach which uses twelve subiterations. In each subiteration the algorithm can use different deleting conditions. This is the main difference compared to fully parallel strategy where the thinning process uses global and predefined delete constraints. Palagyi's algorithm uses subiterations in order to test only specific set of voxels. These sets of voxels are determined with rules called directions. Figure 3 shows twelve directions proposed by Palagyi. Each object boundary voxel in an actual direction can be tested using predefined, for this Subiteration, deleting templates. Palagyi defined 14 deleting templates which are final templates used to test voxels in US direction [9] (see figure 3). Other direction templates are formed by translation of the final US templates using rotation or reflection according to the actual direction rules. Deleting templates can be presented in similar way to Ma and Sonka ones. That is, black dots correspond to object points while white dots represent background points. Other points in template can be both object or background voxels. When all voxels which match deleting templates are marked, the algorithm deletes them and continues to the next subiteration. The algorithm stops if there is no marked voxel for deletion in each subiteration. In that case undeleted voxels form a final skeleton.

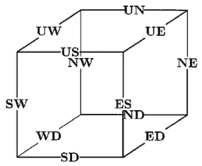

Fig. 3. 12 direction proposed by Palagyi [9].

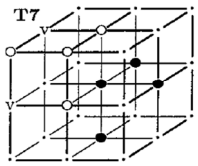

Fig. 4. One of 14 deleting templates assigned to the first subiteration in US direction [9].

A corresponding subiteration based algorithm can be expressed as follows:

Repeat
  For i = 1 *to* 12 *do*
    Mark border points which match the deleting templates predefined for *i-th* direction.
  Delete all marked points
Until no point for deletion in each direction remains

4.3 Voxel Coding Algorithm

The algorithm presented here is complicated and its decryption in details exceeds the article size limitation. Therefore we are going to give only general view of the method. Interesting readers are requested to refer to [14]. The algorithm consists of two steps:
1. Initial skeleton generation
2. Refinement

The first step results in initial skeleton and utilizes a voxel coding scheme – a procedure similar to a discrete minimum distance transform. It uses the coding scheme "n_f-n_e-n_v" ("n_e-n_v" for 2D images) which is described with three integer values greater than 0: n_f, n_e, n_v, ($n_f < n_e < n_v$). First, all the object O voxels are initialized with a code (value) of infinity. Than the propagation starts from seed voxels which are given code 0. Then all the seed F-neighbours, E-neighbours, V-neighbours within an object are given a code of n_f, n_e, n_v, respectively. In the i^{th} iteration, all neighbours of voxels which have been assigned with a code value during the i^{th} -1 iteration are processed. Assume that voxel p is assigned with a value of n for the i^{th}-1 iteration. Thus for the i^{th} iteration all its F-neighbours, E-neighbours, V-neighbours within an object are assigned with value $n + n_f$, $n + n_e$, $n + n_v$, respectively, provided that the new code values are lower than the actual ones (i.e. an infinity value replaced by a code 2, or a code 4 replaced by a code 2). This method prevents voxels coded during an iteration from being coded again in the following one. This coding procedure stops when there is not any voxel to process in the next iteration or the constraint conditions are fulfilled (e.g. a particular voxel is met). The voxel coding procedure applied to 3D (2D) image results in an 3D (2D) image respectively which is called voxel field.

The skeletonization algorithm described in this section utilizes two types of voxel codings:
1. *BS-coding* which uses object boundary voxels as a set of seeds and the generated field is called *BS-field*.
2. *SS-coding* which uses a seed set which consists of only one specific object voxel called *reference point* (RP). The coding results in a field called *SS-field*.

BS-field is utilized to obtain cantered skeleton while SS-field provides useful information about object connectivity and topology. The next step consists in SS-field transformation to a collection of clusters. A *cluster* is defined as a set of connected

object's voxels of the same SS-code; the SS-code is called the *cluster code*. Therefore a cluster can be considered as a connected set of object voxels which belong to the intersection between the object a sphere with center RP and radius equal to the cluster code. During initial skeleton generation one voxel of the highest BS-code is choosen from each cluster to form a centreline. Such a voxel is called *medial point*.

Unfortunately the initial skeleton comprised of medial points, lacks of connectivity therefore the second step - sophisticated refinement, which consists of three steps is utilised. The goal of the refinement first step is to restore connectivity inside branches of the skeleton. Unfortunately the side effects of the first step are folds, branch self-intersections and thick branch fragments. Therefore the second step which smoothes the skeleton has to be applied. Unfortunately the initial skeleton generation procedure gives only information which skeleton branches should be connected but does not connect them. Therefore the third step of refinement is performed which connects extracted branches.

5 Discussion of the Original Algorithm Test Results

The chapter 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 a real human colon image with size 204x132x260 voxels, on standard PC platform computer with Pentium dual core 2GHz processor.

The results are presented in figure 5. Although, FPA produces smooth and traversing a centre of a colon centreline, it also generates some unnecessary branches, loops and flat, one voxel wide surfaces (see figure 5 a). The branches are the result of the algorithm sensitivity to the object shape. The loops are the result of small holes which occur in the colon data because sometimes a colon is so twisted that walls of its two different parts touch each other. In the case a centreline extraction algorithm should not generate a path through the contact area of walls. The last problem (flat surfaces) is the results of inadequate set of deleting templates. If object is thin and flat or in previous iteration the algorithm generated one voxel thin surfaces than proposed deleting templates do 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 produce acceptable centreline for VC.

TSA belongs to the same category of skeletonization algorithms but subiteration strategy produce more sharp and simple skeleton than FPA (see figure 5 b). Although, the algorithm is less sensitive to shape of an object and does not generate many branches and flat surfaces, it detects each hole and generates a loop around it. Therefore the algorithm also needs significant improvements in order to generate high quality centreline for VC.

The last tested algorithm generates the main skeletal path on the first separated step and then extracts its branches. Therefore it is very easy to remove unnecessary branches. It is only a matter to finish the algorithm after the main skeletal path is generated. The result of VC application is presented in the figure 5 c. Unfortunately, two significant drawback of the centreline can be noticed in figure 5 c; losing of

centricity or even touching of the colon walls and passing of the colon wall contact areas. Unfortunately VCA also needs significant improvements but the type of improvements is totally different than in case of TSA and FPA.

Another important aspect of centreline generation algorithms for VC is computational cost which strongly depends not only on the size of an object but also on its shape. The fast centreline generation algorithm can lead to realisation of virtual colon examination immediately after CT scanning of a patient. Working times of tested algorithms for the image presented in figure 2 are included in the table 1. In case of thinning algorithms the time of working strongly depends on the amount of deleted voxels on each iteration. In some specific cases when an algorithm deletes only few voxels on each iteration, the computation time can be very long. FPA is usually faster than TSA. In our tests FPA has been almost 6 times faster than TSA (see table 1). Unfortunately, directional strategy applied in TSA results in the fact that TSA deletes less voxels on each iteration than FPA.

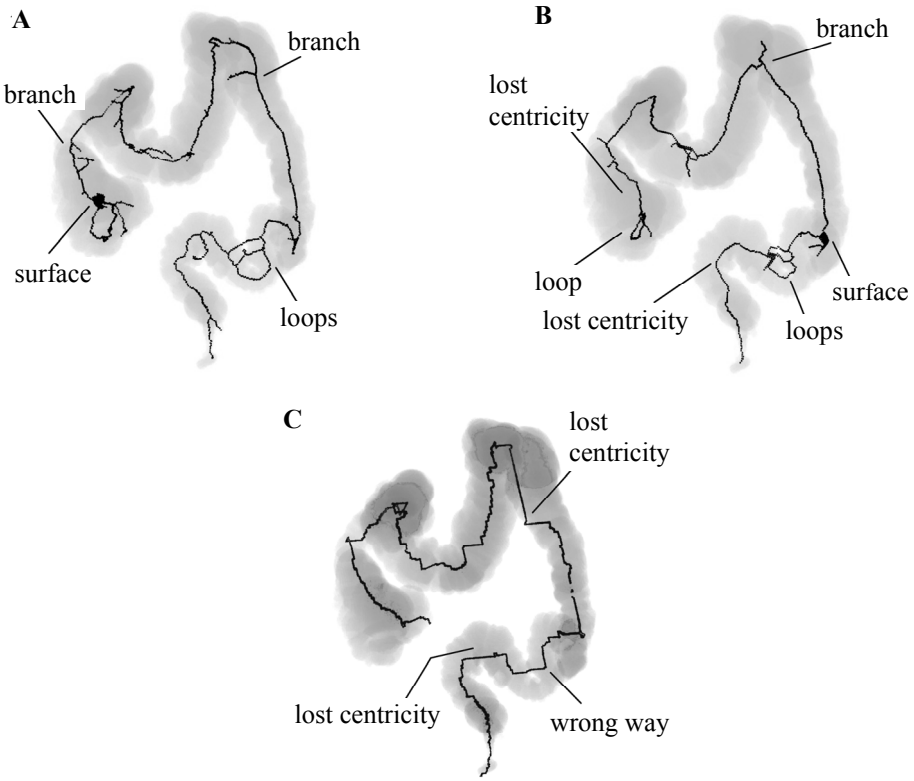

Fig. 5. 3D visualisation of a colon data with a centreline inside represented with dark curve. There are also some special areas indicated which show the interested centreline features. a) Centreline generated with FPA, b) Centreline generated with TSA c) Centreline generated with VCA.

In case of VCA the time of working does not depend so strongly on the features of an object but for complicated objects, with many thin branches, generation of voxel field needs far more time than for simple objects. The most important problem which concerns VCA computational cost is extensive utilisation of connected component labelling procedure (CCL) - a bottleneck of many well known algorithms. The VCA uses CCL many times especially in refinement. This results in the fact that VCA is the slowest algorithm of all tested. Another reason for which VCA works slowly is the fact that the algorithm does not generates unordered set of discrete voxels which form a skeleton, (like PFA and TSA) but it extracts paths of voxels which represents main centreline and branches. Moreover the algorithm return information about structure of the skeleton like starting and ending voxel of each branch, pairs of branches which are connected, length of each branch and so on. The information is useful in further processing of generated skeleton. The authors use it in implementation of the algorithm improvements.

Table 1. Times of working of tested algorithms represented in seconds. The algorithms were examined based on the real colon structure of size 204x132x260 presented in the figure 5.

Algorithm	FPA	TSA	VCA
Total time(sec):	3	17	600

6 Improvements of Tested Algorithms

The chapter presents the author original improvements of the algorithms presented and tested in the previous chapters. As it was shown in previous chapter, the tested original skeletonization algorithms do not generate acceptable centrelines when applied to a real human colon 3D image. Moreover the generated centrelines have serious drawbacks which lead to a conclusion that the algorithms need significant improvements.

In case of FPA and TSA the improvements have been implemented as post processing procedures which analyse skeletons generated by the original algorithms and refine them to obtain acceptable centrelines. Because these two algorithms generate skeletons represented in the same form with similar faults, the authors have implemented and applied one refinement procedure for both algorithms.

The implemented refinements extensively uses a procedure which extracts the *shortest path* (according with the definition of path presented in one of the previous chapters) within the object O, between two voxels p and q from O [14]. The procedure consists of two steps. First the *SSCode* with the q as a seed is extracted until the p voxel is reached. Then in the second step the path is generated starting from p. Assume that on the *i-th* iteration the v_i was added to the path, then on the next iteration the v_i's neighbour with the smallest SSCode is chosen as the next path voxel. The last point of the path must be q since the next point extraction is a code-decreasing process and q has the smallest code in the field.

The whole refinement procedure for FPA and TSA can be divided into three main steps:

1. The aim of the first step is to indicate two ending voxels of a centreline. In order to do so the algorithm generates *SSField* within the skeleton extracted by FPA or TSA, with the seed chosen randomly. The point $e_1$which has the highest SSCode is set as the one of two end points. Then the procedure is used once again starting from the e_1 voxel and the skeleton voxel of highest SSCode is meant to be the second end point e_2.
2. This step aim is to remove all unnecessary branches. To do so the procedure generates shortest path inside a skeleton from e_1 to e_2. The step results in clear unit wide centreline without any branches, surfaces and loops. Unfortunately, this path is wrong if some loops occur in the skeleton. Therefore the path has to be corrected in third step of our refinement procedure.
3. The last step of refinement analyses loops in order to discard false branches which go through the wall connection areas. Indeed, in each loop two paths can be indicated. The longer one which represents entire of a colon and the shorter one which is initially selected in the previous step and trespass the wall connection area. The procedure takes as an input initial skeleton S_{init} generated by FPA or TSA and centreline $C \subset S_{init}$ generated on the step 2 of the refinement. C is a path of voxels which can be represented with the following series: e_1, v_0,...,v_k,...,e_2. On the first iteration the procedure deletes v_0 and attempts to generate the shortest path form e_1 to e_2 inside S_{init}. If the procedure cannot reach e_2, it means that v_0 cannot be deleted. Then the next voxel in C is deleted and the shortest path generation form e_1 to e_2 is performed. Assume that v_k is the first deleted voxel for which the algorithm is able to generate shortest path C_1 from e_1 to e_2 inside S_{init}. It means that there is a loop l in S_{init} which shorter branch b_s was broken by deletion of v_k. Thus C_1 contains longer branch of l which traverses interior of the colon as opposed from b_s which traverses the wall connection area. Therefore, better centreline C_1 replaces the worse C. On the next iteration voxel v_{k-1} from C_1 is deleted and the shortest path is generated form e_1 to e_2 inside S_{init}. The deletion of consecutive voxels and propagation of the shortest paths is repeated until e_2 is the next voxel to delete. The procedure results in the centreline C_{end} obtained in the last successful shortest path extraction. C_{end} does not contain any false branch which goes through a wall connection area.

The VCA improvements concern not only refinement procedure but also the core part. The one of the most serious drawback of the core part of the approach is lack of connectivity between consecutive centreline voxels which significantly complicates refinement. It is a result of the fact that two consecutive centreline voxels v_{i-1} and v_i are chosen from two neighbour clusters c_{i-1}, c_i respectively and have the highest BSCode in the clusters. Unfortunately, the procedure does not guarantee that v_{i-1} and v_i are neighbours. Conducted experiments showed that this lack of connectivity between consecutive voxels occurs very frequently. What is worse the distance between consecutive skeletal voxels very often is amounted to several tens voxels. Such initial skeletons are very difficult to refine and even refined, using original algorithm, do not meet most of the postulates presented in the centreline characteristics section.

Therefore we changed the strategy for choosing the next skeletal voxel. Assume that the voxel v_{i-1} from cluster c_{i-1} was added to the skeleton on the last iteration. Then

on actual iteration from the neighbour cluster c_i of code one less than c_{i-1}'s the voxel v_i which has the highest BSCode is chosen. If distance between v_{i-1} and v_i is less than twenty then v_i becomes the next skeletal voxel. Otherwise the next skeletal point is the closest to v_{i-1}, local maximum of BSField which belongs to c_i. If there is not any local maximum of BSField among c_i voxels then v_i becomes the next skeletal point.

Another modification of VCA concerns refinement procedure which unfortunately is time consuming and does not give good results for complicated objects. Thus the three steps of VCA refinement procedure have been changed into one. The new refinement procedure for each two consecutive non neighbouring, skeletal voxels builds a voxel path which links these two voxels. Assume that on i-th iteration of the procedure voxel v_{i-1} was added to a linking path then on the i-th iteration the next voxel v_i is chosen based on the following formula:

$$F(v_i) = \min_{v \in N(v_{i-1})} F(v) \quad \text{and} \quad \mathrm{F}(\mathrm{v_i}) < \mathrm{F}(\mathrm{v_{i-1}}) \tag{1}$$

$$\text{where}: \ \forall v \in O: \quad F(v) = \frac{SSfield(v)}{BSfield(v)+1} \tag{2}$$

$$N(v_{i-1}) - \text{set of all neighbours of voxel } v_{i-1}$$

If there is no v from $N(v_{i-1})$ for which $F(v) < F(v_{i-1})$ than the voxel from $N(v_{i-1})$ with the minimum *SSCode* is added to the linking path.

Such a refinement procedure is faster than the original one and gives better results in performed tests (see the next section).

7 Test Results of Improved Algorithms

All improved algorithms have been tested on the same volumetric image as original ones. The results are presented in figure 6. Centrelines generated by improved algorithms are definitely better than centrelines generated by their original equivalents. They do not have any branches and loops. What is more, in case of a hole, improved algorithms reject false branch and leave the proper one which traverses the real bulk of the colon. Unfortunately, centrelines generated by improved TSA (ITSA) and improved VCA (IVCA) still suffer from approaching the colon walls especially near holes. What is more centreline extracted with IVCA is not as smooth as others. It is the result of refinement procedure applied in the algorithm which does not guarantee centricity of restored parts of the centreline. Therefore in some areas centreline voxels oscillate around the centre of the colon.

The time of working of improved algorithms applied to the colon image presented in figure 6 are reported in table 2. Comparison of results obtained for improved algorithms and their original equivalents show that refinement procedures significantly extend working time of FPT and TSA. Interesting results were obtained for IVCA which is three times faster than VCA. It is a result of changing refinement procedure to faster one which do not use connected component labelling algorithm.

The new refinement procedure is not only faster but also results in better centreline. In spite of significant acceleration, IVCA is still the slowest algorithm of all tested.

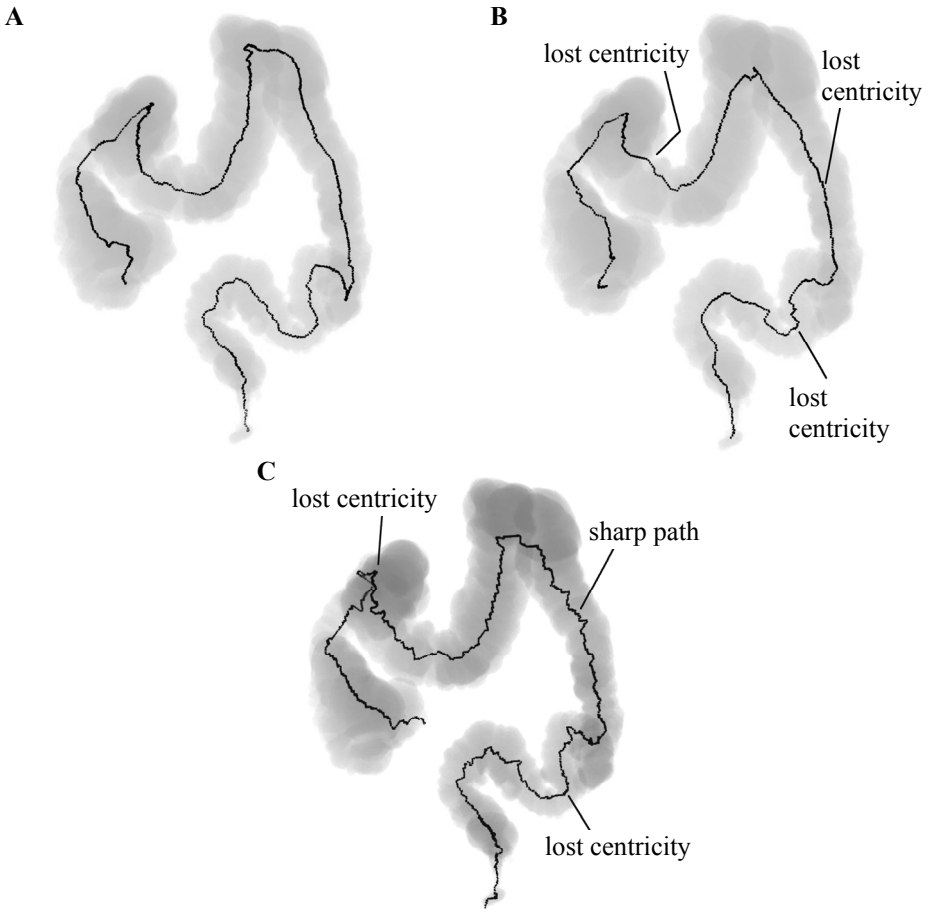

Fig. 6. 3D visualisation of a colon data with a centreline inside represented with dark curve and generated with improved versions of tested algorithms. There are also some special areas indicated which show the interested centreline features. a) Centreline generated with IFPA, b) Centreline generated with ITSA c) Centreline generated with IVCA.

Table 2. Times of working of improved versions of tested algorithms represented in seconds. The algorithms were examined based on the real colon structure of size 204x132x260 presented in the figure 6.

	IFPT	ITSA	IVCA
Total time(sec):	20	35	200

8 Conclusions

In this article three skeletonization algorithms applied to VC were compared. Unfortunately, preliminary examinations showed that original algorithms generate centrelines with many loops and branches. What is more in some areas centrelines are represented by one voxel wide surfaces and have tendency to approach walls of a colon on sharp bends. Therefore the authors have made some improvements to tested algorithms based on their own conceptions. As the result the improved algorithms generate one voxel wide centrelines without any loops and unnecessary branches. The only problem which has not been completely eliminated is local loosing of centricity, in case of centrelines generated with ITSA and IVCA. Finally the best of the tested algorithms is IFPT because it generates centrelines which meet the most postulates in comparison with centrelines generated by other tested algorithms and is fast enough to work on standard PC computer. Unfortunately very promising VCA algorithm has not met the authors' expectations, who have faced a lot of problems with its improvements. The reason for the fact is the wrong assumption that generated clusters approximately represent consecutive cross-sections of the object perpendicular to its centreline which is a basis of the method. The authors have found out that clusters generated on sharp bends of the colon are far away from such cros-sections and are the source of insufficient centreline voxels.

In the future the authors plan to compare more centreline generation methods and made some improvements if necessary to obtain centrelines which meet all postulates presented in the third chapter about centreline characteristic.

References

1. Blum, H.: Transformation for Extracting New Descriptions of Shape. In: Dunn, W. (eds.) Models for the Perception of Speech and Visual Form. Cambridge, Mass. MIT Press, pp. 362--380 (1967)
2. Do Yeon, K., Jong Won, P.: Virtual Angioscopy for Diagnosis of Carotid Artery Stenosis. Journal of KISS: Software and Applications, vol. 30, pp. 821--828 (2003)
3. Guangxiang, J., Lixu, G.: An Automatic and Fast Centreline Extraction Algorithm for Virtual Colonoscopy. 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Shanghai, China (2006)
4. Kong, T.Y., Rosenfeld, A.: Digital Topology: Introduction and Survey. Computer Vision, Graphics, and Image Processing, vol. 48, pp. 357--393 (1989)
5. Ma, C.M., Sonka, M.: A Fully Parallel 3D Thinning Algorithm and Its Applications. Computer Vision and Image Understanding, vol. 64, pp. 420--433 (1996)
6. Ming, W., Zhengrong, L., Qi, K., Lichan, H., Bitter, I., Kaufman, A.: Automatic Centerline Extraction for Virtual Colonoscopy. IEEE Transaction on Medical Imaging, vol. 21, pp. 1450--1460 (2002)
7. Nain, D., Haker, S., Kikinis, R., Grimson, E.L.: An Interactive Virtual Endoscopy Tool. Satellite Workshop at the Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention (2001)
8. Palagyi, K.: A 3-Subiteration 3D Thinning Algorithm For Extracting Medial Surfaces. Pattern Recognition Letters, vol. 23, pp. 663--675 (2002)

9. Palagyi, K., Kuba, A.: A Parallel 3D 12-Subiteration Thinning Algorithm. Graphical Models and Image Processing, vol. 61, pp. 199--221 (1999)
10. Palagyi, K., Kuba, A.: A Parallel 12-Subiteration 3D Thinning Algorithm to Extract Medial Lines. 7th Computer Analysis of Images and Patterns, pp. 400--407 (1997)
11. Saha, P.K., Chaudhuri, B.B.: 3D Digital Topology Under Binary Transformation with Applications. Computer Vision and Image Understanding, vol. 63, pp. 418--429 (1996)
12. Wang, T., Basu, A.: A Note on a Fully Parallel 3D Thinning Algorithm and Its Applications. Pattern Recognition Letters, vol. 28, pp. 501--506 (2007)
13. Wegenkittl, R., Vilanova, A., Hegedust, B., Wagner, D., Freund, M.C., Groller, E.M.: Mastering Interactive Virtual Bronchioscopy on a Low-End PC. Proceedings Visualization 2000, Salt Lake City, USA, pp. 461--464 (2000)
14. Zhou, Y., Toga, A.W.: Efficient Skeletonization of Volumetric Object. IEEE Transactions on Visualization and Computer Graphics, vol. 5, pp. 196--209 (1999)