A tree-branch searching, multiresolution approach to skeletonization for virtual endoscopy

Dongqing Chen*, Bin Li, Zhengrong Liang, Ming Wan, Arie Kaufman, and Mark Wax
Departments of Radiology and Computer Science, SUNY, Stony Brook, NY 11794

ABSTRACT

One of the most important tasks for virtual endoscopy is path planning for viewing the lumen of hollow organs. For geometry complex objects, for example the lungs, it remains an unsolved problem. While alternative visualization modes have been proposed, for example, cutting and flattening the hollow wall, a skeleton of the lumen is still necessary as a reference for the cutting. A general-purpose skeletonization algorithm often generates redundant skeletons because of the local shape variation. In this study, a multistage skeletonization method for tree-like volumes, such as airway system, blood vessels, and colon, was presented. By appropriately defining the distance between voxels, the distance to the root from each voxel in the volume can be effectively determined with means of region growing techniques. The end points of all branches and the shortest path from each end point to the root can be extracted based on this distance map. A post-processing algorithm is applied to the shortest paths to remove redundant ones and to centralize the remained ones. The skeleton generated is one-voxel wide, along which every branch of the "tree" can be viewed. For effectively processing volume of large size, a modified multiresolution analysis was also developed to scale down the binary segmented volume. Tests on airway, vessel, and colon dataset were promising.

Keywords: skeletonization, tree-branch searching, region growing, multiresolution, and virtual endoscopy.

1. INTRODUCTION

In recent years, virtual endoscopy (VE) has been widely studied as a non-invasive method for detection of epithelial tumor or lesion in various kinds of hollow organ [5,6,15,16,20]. With this technology, the three-dimensional (3D) model of the inner surface of the hollow organ can be constructed from the computed tomography (CT) or magnetic resonance (MR) cross-sectional images. The surface model can then be viewed with different display modes. Although there is no comparative clinical trial revealing the advantages of VE on reliability and specificity over conventional endoscopy (CE), it has been demonstrated that VE is patient comfortable, cost-effective, free of risks, and capable for viewing all locations in the entire lumen where CE may have limited views.

For geometry complex organs, like the cardiovascular system, the pulmonary tree, and the colon, an appropriate way to view the entire lumen must be carefully designed. Earlier work has dedicated to calculate a fly-through path along which the virtual camera can move from one end to the other [3,9,10,17]. In those studies, the smoothness and the centralization of the line are the basic requirements. Instead of navigation inside the lumen, several alternative visualization modes for virtual colonoscopy have been presented in recent years [19,21,22]. The colon was bisected or flattened and the resulted local three-dimensional (3D) views were displayed sequentially. Though no fly-path was needed there, a thin skeleton of the lumen was still required as a reference when the colon was cut or deformed. In virtual bronchoscopy and angiography, the geometry structures of the airway and the vessel trees are much more complicated. A thin skeleton that is helpful for accurately describing the tree structure is highly desired [8,23]. From the discussions above, one may conclude that no matter what purpose is pursued on, and no matter what name is called, an abstract geometry object inside the lumen is desirable to implement VE on geometry complex organs. This study is dedicated to develop a general-purpose method to generate this abstract object for tree-like 3D volume datasets. Considering the generality, we call the abstract object as skeleton. It is obvious that most of the cavity human organ can be abstracted as a tree structure.

Skeletonization has been studied for decades in pattern recognition and computer vision. The definition of skeleton is application-driven. Hence, flexible skeletonization is desired in various applications [4]. In VE, the skeleton is used as both a shape descriptor and a guidance of viewing. As a shape descriptor, the skeleton is usually generated by thinning algorithm [11,13]. Although the thinning algorithms generate good skeleton for tree-like volume dataset, for example the coronary arterial tree [8], they do not always preserve the homotopy and might create redundant branches for colon lumen due to the

* For further information: E-mail: dchen@clio.rad.sunysb.edu; Tel: (631) 444-2508; Fax: (631) 444-6450.

haustral folds. For application-specific thinning, the distance transform could be applied to create a roughly central skeleton [10,24]. However, if the skeleton is to be used as a navigation fly-path for a virtual camera, it has to be refined by other techniques [3,9,10,17]. In the latter situation for navigation application, the geometry knowledge of the organ has to be known prior to the path-planning, for example, the start and the target points located at the two ends of a tubular volume must be provided. By this philosophy, automatic analysis to geometry structure is difficult.

Wood and coworkers developed a multi-stages scheme to generate skeleton for lung tree structure [23] and they measured the 3D structure based on the skeleton. The initial skeleton was determined by a region growing algorithm. While growing from a seed voxel, voxels in a 26-connected volume were iteratively assigned with an integer number that was not overwritten in subsequent iterations. By weighting each of the voxels grouped with the same integer descriptor number equally, and averaging the x-, y- and z-coordinates of each of the voxels comprising the group, a centriod element is calculated for each grouping of voxels. The algorithm then connected any two centroid elements with a line. Post-processing steps was needed to refine the initial path to a continuous skeleton. Although the region growing algorithm was implemented much faster than other skeletonization algorithm, there were several limitations in their method. After the number assignment, the voxels assigned with the same number but located at different branches must be grouped separately. This is necessary for determining the centroids by averaging. The grouping procedure will take extra time and the grouping algorithm was not presented in detail in [23]. On the other hand, the centroids-connecting procedure is complicated at the bifurcation, especially at bifurcation with more than 2 branches, and some kinds of branch point are needed to be repositioned manually with a specifically designed central-axis editor. Blezek and Robb introduced a more accurate integer distance assignment algorithm for virtual endoscopy [2]. They focused on generating the shortest path for a given end, where the well-known chamfer distance was utilized. However they did not discuss automatic structure analysis.

In this work, we deal with 3D 26-connected tree-like volume. A root voxel is assumed to be known. This assumption is reasonable in VE. The presented skeletonization method is a multistage one. At the first step, the algorithm generates a one-voxel-wide, 26-connected skeleton tree with root at the given root voxel. The algorithm is based on region growing technique, but with a different way for number assignment. Considering this difference, our method is some kind of extension of the previous work [2,23]. The second step is to refine the skeleton tree for a specific application. The algorithm is unique in that the number of branches of the tree can be automatically detected and the definition of the end of the branch is flexible to many applications. The algorithm is effective in implementation. In order to process volume dataset of large data size, like the colon lumen, a multi-resolution implementation scheme is developed. Our method can be divided into three parts. The first part describes the branch-searching algorithm based on the region growing techniques. The second part is a skeleton refining method. The last part presents a speed-up algorithm for large volume datasets based on multiresolution analysis. The method has been tested on colon, airway, and vessel tree datasets. The results are promising. The advantages and limitations of the method will be discussed later.

2. MEDTHODS

2.1 Definition of the skeleton

We focus on a tree-like, 26-connected (also called point connected) [12], 3D volume with a single root but possible multiple branches. The input data is a stack of binary images of the same size with voxels in the volume labeled by *1* and voxels of the background labeled by *0*. The root voxel is assumed to be known. The **skeleton** is defined as a subset of the volume with the following attributes: 1) It preserves the homotopy of the tree; 2) It is 26-connected; 3) It is on one voxel thick; 4) It approximates the central axes of the branches; and 5) It is as smooth as possible. In VE, the last 4 features of the skeleton have been discussed extensively in [17]. The first requirement has been relatively less discussed. It is difficult to provide a quantitative measurement for the preservation of the homotopy, because it is an application-specific term in VE. For example, the skeleton for the colon is desired to be a single curve between two ends and any branch, even though there are many haustral folds on it. On the other hand, when we deal with pulmonary arterial-tree, a small shoot with several voxel-length may be a real branch. In this study, we will develop a skeletonization method that is flexible to this kind of specific application requirement. Furthermore, the number of branch should be automatically detected during the skeletonization.

2.2. Calculation of the distance map

The basic idea of our method is based on the fact that there is a unique end point on each branch, which has the local longest distance to the root of the tree. Although in continuous space there exists unique shortest path connecting two points inside the volume, there may be many shortest path connecting two fixed voxels in discrete space due to different kind of definition

of distance [12]. There are many approaches to define the distance in the voxel space [12]. In this study, the distance from each voxel to the root is to be calculated and this integer value of distance will be assigned to each voxel. The result is called a *distance map* to the root. Considering the effectiveness in implementation, the seeded region growing algorithm may be the best one for calculating the distance map since it goes through each voxel in a single round. Of course, different definition of the distance would generate different distance maps.

Wood et al. developed a method to determine a distance map from the region growing algorithm by assigning an integer distance descriptor to each voxel in the volume [23]. The seed point was assigned a value of *0*, whereas all 26-connected neighbors of the seed, which was inside the volume, were assigned a value of *1*. The assignment of the *1*'s completed the first iteration. The algorithm then assigned those voxels 26-connected to a *1* the integer value of *2* to complete the second iteration. At the k-th iteration, all voxels 26-connected to a *k-1* were assigned the integer value of *k*. The region growing algorithm is independent of orientation, since contiguous voxels were added within the 3D cube. However, one could not find the end of branch with this distance map since there might be an end surface, not a point, with the same integer value. The left drawing of Fig. 1 shows a two-dimensional (2D) case of this situation.

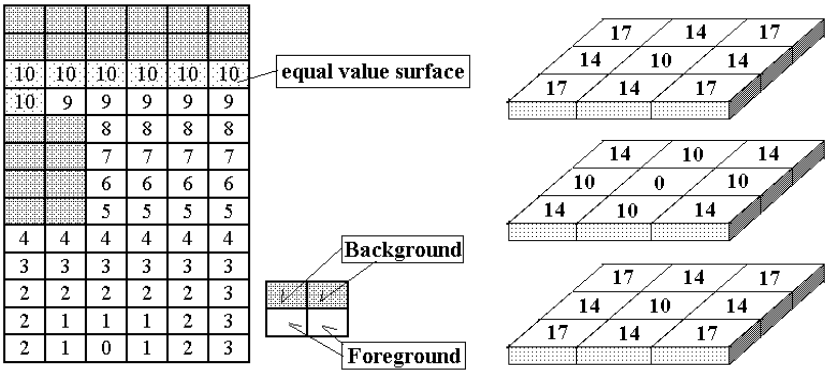


Figure 1. A 2D example of a distance map with uniform distance assignment is showed on the left. An equal value surface (brighter shade area with value 10) appears on the single branch end. In the right drawing, the integer plate for distance assignment is shows.

To overcome this problem, we modified the method of distance descriptor assignment. Instead of assigning the same value of integer at the k-th iteration, the 26-connected neighbors in the volume of those assigned voxel at the last iteration will be assigned with a different integer value according to a cubic distance plate (right of the Fig. 1). Those number in the plate are integer approximation of the Euclidean distance between a voxel and its 26-connected neighbors. The reason of using the approximation is to reduce the possibility of the appearance of multiple shortest paths between two voxels and the appearance of equal-value surface. There are only three kinds of Euclidean distance from the neighbor to itself. These are $1, \sqrt{2} \approx 1.4,$ and $\sqrt{3} \approx 1.7$. It is reasonable to approximate them with integer 10, 14, and 17 in order to save implementation time. This distance assignment is less accurate than the chamfer distance, but it performs well in all our experiments. The new algorithm to calculate the distance map is described in the following.

1) Label root voxel with integer 0;
2) Construct a queue and line up the root in the queue;
3) **If**(There is at least one voxel in the queue)
   Serve the voxel **x** on the top of the queue:
   For(each of **x**'s 26-connected neighbor voxel **y**) {
   If(**y** in the volume and has not been labeled yet) {
   Line up the **y** in the queue;

   /* label the voxel **y** */
   Set $dist = 999999$;
   For(each of **y**'s 26-connected neighbor voxel **z**) {
   If(**z** in the volume and has been already labeled with an integer of n_z) {
   $d_z \equiv n_z + d(y,z)$;
   where $d(y,z)$ is 10, 14 or 17 if the Euclidean distance between **y** and **z** is

$1, \sqrt{2}$, or $\sqrt{3}$, respectively;
}
If($dist > d_z$) {
label **y** with integer $dist$;
$dist = d_z$;
}
}
}
}
x leaves the queue;
}
Else {
end of calculating the distance map.
}

The above algorithm is a typical region growing algorithm. Voxels in the queue is served according to the first-in and first-out rule. The resulted integer for each voxel in the volume is regarded as the distance from itself to the root. The algorithm is different from that of [23] on the integer assignment strategies. Here, the number is assigned to a voxel to make sure that it is the shortest distance to the root. *The most important feature of the generated distance map is that each voxel, except the root, in the volume has at least one 26-connected neighbor in the volume whose distance to the root is less than that of this voxel.*

Of course, the distance map generated by the algorithm is not unique. In the labeling procedure, different checking order of those 26-connected neighbors will create different distance maps. Although this may lead to different locations of the end point for each branch of the tree, it does not change the fact that there is at least one farthest point for each branch. This means that the generated distance map provides enough information for tree branch structure.

2.3. Determination of the ends of branch

From the feature of the distance map we know that each end of branch of the volume dataset must possess a local maximum distance to the root. Hence, to determine the ends is just to find out all the local maximum voxels. The maximum distance is a relative term. One has to set a limit subspace to compare the distances. In VE, different organs may possess different feature sizes. For a specific application problem, there may be a minimum length and a maximum width for all branches that have clinical significance. The voxel length associated to these sizes can be obtained based on the imaging parameters. A 3D discrete voxel ball with radius associated to these sizes can then be determined. Using this ball plate, each voxel in the volume will be checked whether its distance is the maximum within the ball with the center at that voxel. If yes, it is an end voxel; If not, check next voxel. After going through all the voxels in the volume, both the number of branches and the end voxels of the tree-like volume are obtained simultaneously. This is the unique advantage of our method over those of others.

It must be note that if the radius of the plate ball is too large, it may take much time for checking procedure. If the radius is too small, there may be redundant branches appeared. On the other hand, for some specific application like the colonoscopy, one knows that it is a tubular volume with only one branch. Hence, the branch searching is just searching the maximum distance in the distance map. This will greatly reduce the searching time.

2.4. Searching for the shortest paths

For each end, a shortest voxel path from the end to the root will be generated. All the shortest paths are generated one by one from the farthest end to the nearest end. For the farthest end, the searching algorithm begins at the end. Then the voxel of one of its 26-connected neighbors that realizes the minimum distance within all those neighbors is chosen as the second voxel in the path. The searching step recursively applies to the latest voxel chosen in the path until meets the root. All the chosen voxels constitutes of a one-voxel thick path from the farthest end to the root. For the remaining ends, the searching procedure is similar, except the stopping rule. It stops when it meets a voxel that has already been chosen as a path voxel for previous ends.

A rough skeleton of the tree can be obtained by collecting all those shortest paths. From the generating procedure we know that the skeleton is a one-voxel thick, 26-connected tree with the same number of branch. A problem must be considered. If a

branch is a sub-branch of his father branch, the shortest path of the son may not merge into the shortest path of his father before reaching the root. This situation frequently occurs when the tree trunk is too short. It will cause a little bit difficulty to determine the bifurcation of the tree in this case. At present, we have not overcome this problem completely. We applied a sorting algorithm prior to the shortest paths searching. The end points were sorted from the farthest to the nearest at first. Then, the shortest paths were generated one by one from the farthest to the nearest end points. Our numerical experiments showed that this reduced the possibility to false connection between father and son branches.

2.5. Refinement of the rough skeleton

The requirements on the skeleton are application specific as mentioned before. If the skeleton is used to represent the tree structure, a rough skeleton is good enough for the purpose. More exactly, the rough skeleton satisfies the conditions 1), 2), and 3) in the definition of the skeleton. To ensure a skeleton that satisfies all those 5 conditions, a refining algorithm is needed and presented in the following as a post-processing step.

First, we manage to centralize the skeleton. As stated previously, a rough skeleton is generated branch by branch from the longest to the shortest end. The centralization procedure is also applied branch by branch from the longest branch to the shortest branch.

For the longest branch path, some voxels in the path are chosen with a uniform interval between the farthest end and the root. The interval is set to be 4 to 8 voxels in this study. The tangent direction on the chosen voxel is then calculated. A plane crossing each chosen voxel that is perpendicular to the tangent can be determined. A 2D area of the intersection between the volume and the plane is created. The center voxel of this 2D area is computed by the maximum disk technique [7]. This center is the centralization result of the chosen voxel. All the centralized centers and the two ends are finally connected by the well-known bi-cubic, B-spline interpolation [18].

For the remaining branch paths, the centralization procedure is similar, except the determination of two ends. One end is the branch end which is fixed. The other end was recorded in the branch searching stage. It has to be replaced by the nearest voxel which is located in the previously centralized paths. After the two end being set, the following centralization step is the same as that mentioned in the last previous paragraph.

2.6. Multiresolution approach

For some application the volume data has a very large size. For example, in colonoscopy, there are often millions of voxels in a single colon lumen. Processing directly on the large volume is time costly. Noting that the tree structure can be preserved within a range of scales, the large volume can be shrunk to a smaller scale space for structure analysis. This is similar to the situation that one has to obtain an overview of a large object from a distance.

We present a shrinking method based on multiresolution analysis theory [24]. The input data is a stack of binary images with the same size. The x-direction is along slice image width, the y-direction is along the slice image height, and the z-direction is along the direction of slice by slice. The foreground voxels in the tree volume are set to value 128 and the background voxels are set to value 0. The Daubechies' bi-orthogonal wavelet with all rational coefficients [1] is employed. The one-dimensional (1D) discrete wavelet transformation (DWT) is first applied along to the x-direction row by row. Only the lower frequency components are retained and packed. The computation is implemented in floating points. Noting that the DWT is applied to the binary signal, there are two kinds of nonzero coefficients in the lower frequency component. One kind is of value 128 and this kind of coefficients locates at the interior of the volume. Another kind is of value not equal to 128 and this kind locates at the boundary of the volume. Fig. 2 shows a 2D example. Before progressing to the next step, coefficients of the second kind are thresholded. If its absolute value is larger than a pre-set threshold **T1**, the value of the coefficient is set to 128; otherwise, it is set to 0. The result is a stack of binary images with half row size of the original dataset. Then, the same DWT is applied to the resulted dataset along the y-direction column by column, where the similar thresholding is employed to the lower frequency components. The result is still a stack of binary images but with both half row and column size of the original dataset. Finally, the DWT is applied to the last result along the z-direction and the lower frequency components are retained. This step completes the first level decomposition.

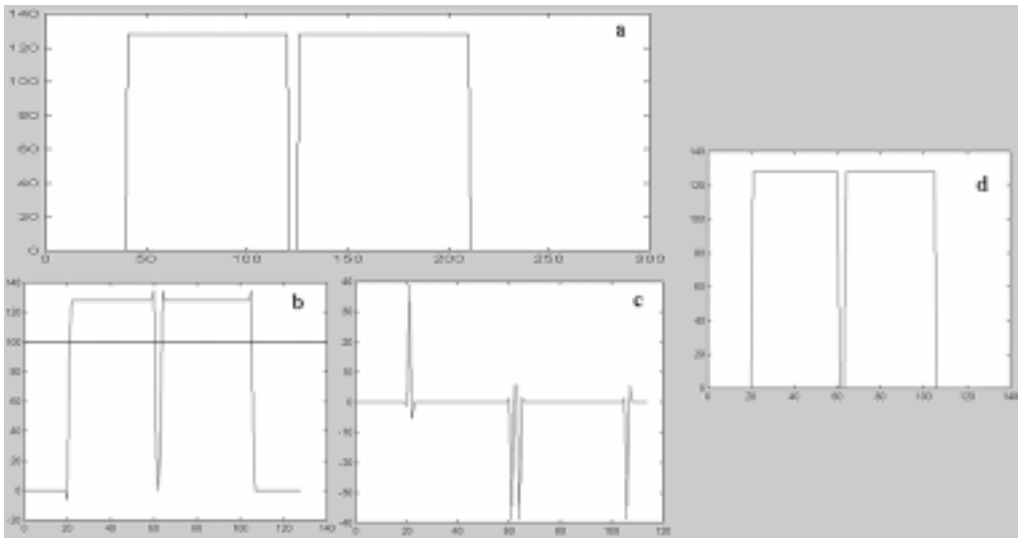


Figure 2. This figure demonstrates the 1D DWT thresholding on binary image. The input binary signal is depicted in (a). The low (b) and high (c) frequency components in the result of DWT are displayed. The line at the value 100 in (b) shows the threshold value. The thresholding result is depicted in (d).

The resulted dataset of the first level decomposition is of half size in all three directions as comparing to the original dataset. If the shrinking procedure stops at this level, the finial thresholding is applied. It revalues those coefficients of nonzero and non-128 value. If the absolute value of this kind of coefficient is larger than a pre-set threshold **T2**, it will be revalued as 128; otherwise, it is revalued as 0. If further shrinking is needed, the same thresholding algorithm is applied with the threshold **T1**. Further shrinking is the same as the previous one, but is applied to the result shrunk at the last previous level. The decomposition procedure can be recursively applied until the resulted volume meets the requirement. In VE, the slice images are of 512×512 pixel size. The maximum decomposition level is usually three.

The volume is isotropicly shrank in all directions with the presented method. Those two pre-set thresholds **T1** and **T2** are used to control the degree of shrinking. If the volume is over shrunk, connectivity may be lost in the shrunk volume. If it is less shrunk, two separate branches may merge into one branch in the volume shrank. The larger the two thresholds, the thinner the shrunk volume is. The range of those two thresholds is $[0, r \times 128]$, where 0<**r**<1. From our numerical experience, the appropriate range for VE is $r \in (0.08, 0.28)$ for **T1** and $r \in (0.7, 0.98)$ for **T2**. The exactly determination is depended on the feature size of practical application. It is desirable that the structure information is still retained in the shrunk volume.

After shrinking the original volume, the same tree branch searching procedure can be applied to the smaller volume or more complicated skeleton generating method may be employed. The resulted skeleton of smaller scale can then be mapped back into the original scale space. The image of the smaller scale skeleton is no longer a connected path in the original scale space. These voxels in the image are the control points for the finial skeleton. The control points are first centralized using the same algorithm as described previously, and then, they are interpolated to form the finial skeleton.

3. RESULTS

The presented method was tested on three kinds of dataset: the colon lumen, the major airway, and the abdominal aorta. One case from each kind of dataset is showed here. All the datasets were extracted from CT images of 512×512 matrix size. The slice numbers are 389, 202, and 221 for colon, airway, and aorta, respectively. The numbers of voxels in each lumen are showed in Tab. 1. The implementation times for processing those three cases are also listed. The time was counted in seconds. The numbers of branches listed were detected automatically using the algorithm. For the colon case, the branch searching algorithm is set to find out the maximum distance. The algorithm was implemented in C code on a SGI /OCTANE workstation. No parallel processing technique was employed. It is equipped with dual 195MHz, MIPS R10000 CPU, MIPS R10010 FPU. The main memory is 896 Mbytes. The installed operating system is IRIX64 Release 6.5.

Fig. 3 depicts the skeletons generated for the colon case where the colon is displayed transparently and is viewed from two different directions. Fig. 4 shows a human airway and the generated skeleton. In Fig. 5, the skeleton generation procedure for

an abdominal aorta is displayed. The initial skeleton and the generated final skeleton are showed together with the original aorta. The multi-resolution approach method was tested for a colon lumen (Fig.6). Although there is a segment of small bowel attached to the colon lumen, the generated centerline is not affected by the attachement. The implementation time for this case is 146 seconds. The time is longer than that of listed in Tab. 1. The reason is that the range searching for local maximal distance is replaced by the searching of the farthest point. All the results are promising.

Table 1. The implementation time and the range for end searching

	Implementation time (s)	Number of branch	Number of voxel	Range for end searching
airway	14	13	41,529	6
colon	36	1	3,090,526	None Apply
aorta	48	29	235,161	8

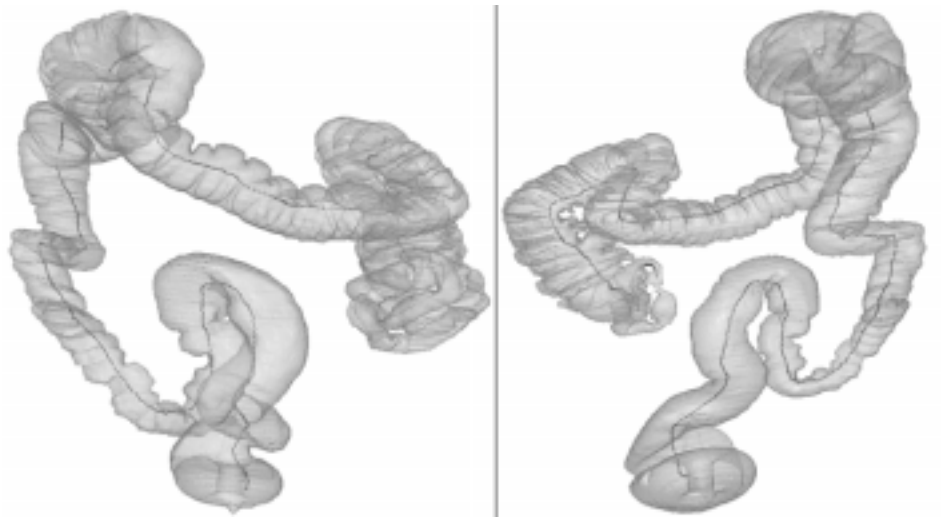


Figure 3. The centerline generated for a colon is viewed from two different directions.

Figure 4. A major airway (left) and the generated skeleton (right).

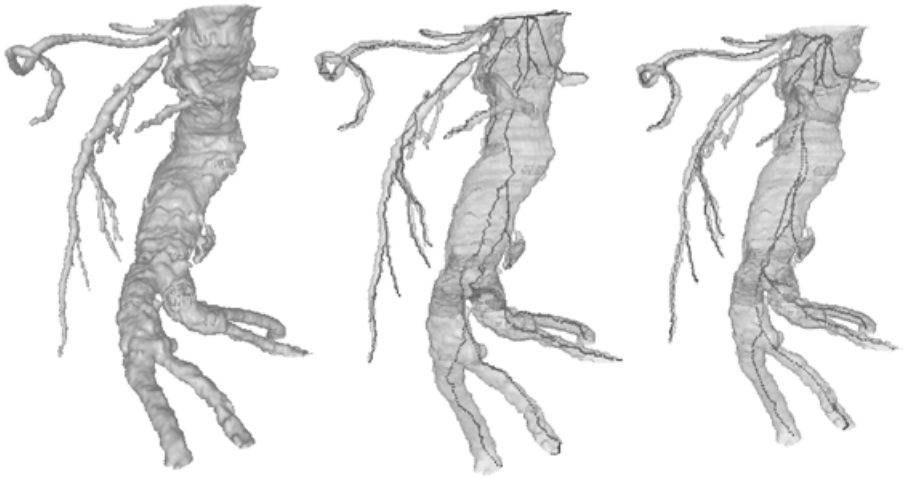
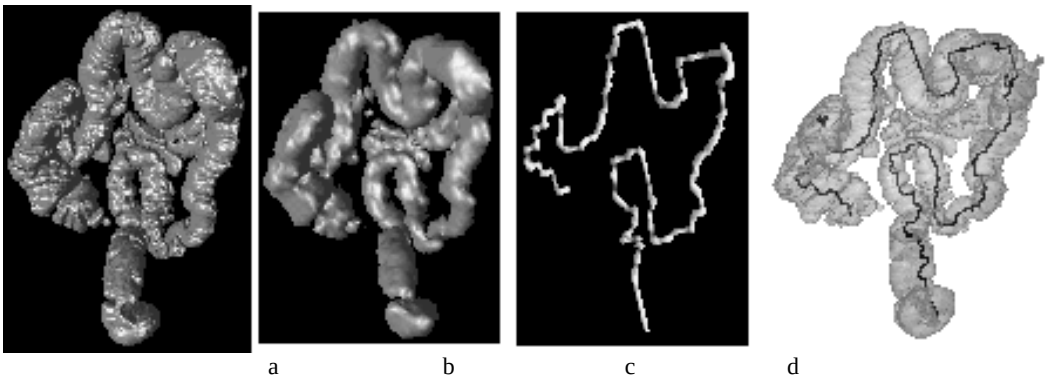


Figure 5. The left picture is the abdominal aorta. The center picture is the initial skeleton for the aorta tree, and the right picture is the centralized final skeleton, where the aorta is displayed transparently.



a b c d

Figure 6. The depiction of the multiresolution approach. The colon lumen (a) is first shrank into a smaller size (b). The centerline of the smaller volume is calculated (c). Then the centerline of the smaller volume is mapped back to original volume. The image of the map is finally centralized and interpolated to produce the finial centerline for the original colon lumen (d).

4. DISCUSSION

Our method performed satisfactorily. There may be some limitations with this method. First, the skeleton is located approximately at the center of the branch. The distance map can reduce the possibility of non-uniqueness for the shortest path, but it does not help the centeralization of the skeleton. The root usually should be set at the center of the bottom area. This will make the shortest path automatically locating at the center of the trunk, but this is not the case for other branches. Second, the range for searching the ends of branches depends on both the length of the branch and the width of the branch. To choose a satisfactory range one must know the accurate information about the size of the branch. This may not be easy for some application, such as in the angiographies, where the vessel tree is complicated. Third, search of bifurcation and measurement of the angle between branches based on our skeleton are not discussed in this paper. This is a possible future work.

In conclusion, the approach of tree branch searching is a feasible method to integrate the skeleton generation and the tree-structure analysis. The developed seeded region growing techniques for this purpose are computationally efficient. The scaling-transformation method greatly reduces the skeleton generation time for large datasets. The method is suitable to both fly-path planning for virtual endoscopy and geometry structure analysis. From the perspective of general-purpose skeletenization view, the tree-branch searching approach has the potential application to extract simple thin skeletons from a more complicated shape. All the steps are implemented automatically, except the determination of the root.

ACKNOWLEDGEMENTS

This study was supported by NIH Grant #CA79180, Grant #NS33853, Grant #HL51466, and EI Awards of AHA.

The authors are grateful to the valuable comments of Drs. Lichan Hong, Taosong He, and Perry Horwich.

REFERENCES

1. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Trans, on Image Processing*, vol. 1, pp. 205-220, 1992
2. D. Blezek and R. Robb, "Centerline algorithm for virtual endoscopy based on chamfer distance transform and Dijkstra's single source shortest path algorithm," in *Physiology and Function from Multidimensional Images*, Chin-Tu Chen, Anne V. Clough, Editors, Proceedings of SPIE vol. 3660, pp. 225-233, 1999
3. R. Chiou, A. Kaufman, Z. Liang, L. Hong, and M. Achniotou, "Interactive path planning for virtual endoscopy," *IEEE Trans. On Nuclear Science*, 1999
4. M. Dsesilligny, G. Stamon, and C. Suen, "Veinerization: A new shape description for flexible skeletonization," *IEEE Trans. On PAMI*, vol. 20, pp. 505-521, 1998
5. H. M. Fenlon and J. T. Ferrucci, "Virtual colonoscopy: What will the issues be?" *AJR* vol. 69, pp. 453-458, 1997
6. M. P. Fried, V. M. Moharir, H. Shinmoto, A. M. Alyassin, W. E. Lorensen, L. Hsu, and R. Kikinis, "Virtual laryngoscopy," *Ann. Otol Rhinol Laryngol.*, vol. 108, pp. 221-226, 1999
7. Y. Ge and M. Fitzpatrick, "On the generation of skeletons from discrete Euclidean distance maps," *IEEE Trans. On PAMI*, vol. 18, pp. 1055-1066, 1996
8. W. Higgins, W. Spyra, R. Karwoski, and E. Ritman, "System for analyzing high-resolution three-dimensional coronary angiograms," *IEEE Trans. On Medical Imaging*, vol. 15, pp. 377-385, 1996
9. L. Hong, S. Muraki, A. Kaufman, D. Bartz, T. He, "Virtual voyage: Interactive navigation in the human colon," *Computer Graphics Proceedings*, Ann. Conf. Series, pp. 27-34, 1997
10. L. Hong, Z. Liang, A. Viswambharan, A. Kaufman, and M. Wax, "Reconstruction and visualization of 3D models of colonic surface," *IEEE Trans. On Nuclear Science*, vol. 44, pp. 1279-1302, 1997
11. S. Lobregt, W. Verneek, and F. Groen, "Three-dimensional skeletonization: principle and algorithm," *IEEE Trans. On PAMI*, vol. 2, pp. 75-77, 1980
12. G. Lohmann, *Volumetric Image Analysis*, Wiley and Teubner, 1998
13. C. Ma and M. Sonka, "A fully parallel 3D thinning algorithm and its applications," *Comput. Vision and Image Understand*, vol. 64, pp. 420-433, 1996
14. S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L_2(R)$," *Trans. Amer. Math. Soc.*, vol. 315, pp. 69-87, 1989
15. E. M. Merkle, A. Wunderlich, A. J. Aschoff, N. Rilinger, J. Gorich, R. Bachor, H. W. Gottfried, R. Sokiranski, T. R. Fleiter, H-J Brambs, "Virtual cystoscopy based on helical CT scan datasets: perspectives and limitations," *The British journal of Radiology*, vol. 71, pp. 62-267, 1998
16. K. Mori, Y. Suenaga, J. Toriwaki, J. Hasegawa, H. Anno, K. Katada, and H. Natori, "Automated display of anatomical name of branchial branches in virtual bronchoscopy system and its application as a training tool for medical students," in *Physiology and Function from Multidimensional Images*, Chin-Tu Chen, Anne V. Clough, Editors, Proceedings of SPIE vol. 3660, pp. 301-312, 1999
17. D. S. Paik, C. F. Beaulieu, R. B. Jeffrey, G. D. Rubin, and S. Napel, "Automated flight path planning for endoscopy," *Med. Phys.* Vol. 25, pp. 629-637, 1998
18. W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipesin C: The Art of Scientific Computing*, the second edition, Cambridge Univ. Press, 1992
19. J. Reed and C. D. Johnson, " Virtual pathology: a new paradigm for interpretation of computed tomographic colonography," in *Image Display*, Proc. of SPIE, vol. 3335, 1997
20. D. J. Vining, "Virtual endoscopy: Is it reality?" *Radiology*, vol. 200, pp. 30-31, 1996
21. G. Wang, S. Dave, B. Brown, Z. Zhang, E. McFarland, J. Haller, "Colon unraveling based on electrical field: Recent progress and further work," in *Physiology and Function from Multidimensional Images*, Chin-Tu Chen, Anne V. Clough, Editors, Proceedings of SPIE vol. 3660, pp. 125-132, 1999
22. G. Wang, E. G. McFarland, B. P. Brown, and M. W. Vannier, "GI tract unraveling with curved cross section," *IEEE Trans. On Medical Imaging*, vol. 17, pp.318-322, 1998
23. S. Wood, E. Zerhouni, J. Hoford, E. Hoffman, and W. Mitzner, "Measurement of three-dimensional lung structures by using computed tomography," *J. Appl. Physiology*, vol. 79, pp. 1687-1697, 1995
24. Y. Zhou, A. Kaufman, and W. Toga, "Three-dimensional skeleton and centerline generation based on an approximate minimum distance field," *Visual Computer*, vol. 14, pp. 303-314, 1998