Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

An Automatic and Fast Centerline Extraction Algorithm for Virtual Colonoscopy

Guangxiang Jiang[1], Lixu Gu[1]
[1]Computer Science, Shanghai Jiaotong University, P.R.China

Abstract-This paper introduces a new refined centerline extraction algorithm, which is based on and significantly improved from distance mapping algorithms. The new approach include three major parts: employing a colon segmentation method; designing and realizing a fast Euclidean Transform algorithm and inducting boundary voxels cutting (BVC) approach. The main contribution is the BVC processing, which greatly speeds up the Dijkstra algorithm and improves the whole performance of the new algorithm. Experimental results demonstrate that the new centerline algorithm was more efficient and accurate comparing with existing algorithms.

I. INTRODUCTION

With the development of computer graphics and medical image technology, virtual endoscopy (VE) system has become a promising computer-assistant clinic application. Typically, a VE system builds a 3D virtual organs in computers from 2D Computed Tomography (CT) or Magnetic Resonance Image (MRI) image slices, provide a interactive interface which allows the users(doctors) to look around the organs. Comparing to traditional endoscopy system, VE has many advantages such as non-invasive, and it can deal with some special organs which are impossible to be accessed using a real endoscopy (e.g. blood vessels). In recent years, many VE systems have been proposed in different fields, For example, virtual colonoscopy (VC) [1], virtual Bronchoscope [2] and Virtual Angioscopy. Some of them have also been applied into clinic trail.

In the VE system, navigation is one of the most important models, which provides the interactive user interface [3]. At the beginning, many systems' navigation interface is pure manual. Then, path planning was proposed and the path was generated from some key points by interpolation, but the key points were still human-selected. Recently, many centerline algorithms are invented to extract the centerline of hollow organ automatically. The centerline algorithms are more effective and user-friendly than former methods, but typically they are still time consuming, particularly when the image data is larger.

In this paper, we introduce a novel colonoscopy centerline extraction algorithm and prove that it improves the existing distance mapping algorithm significantly and achieves a near-real-time performance.

The architecture of the paper is organized as following: in part II, we firstly give a brief review of the basic centerline extraction theory and then expatiate the proposed algorithm. In part III, we test the new algorithm by two datasets and compare it with the traditional algorithms.

II. METHODOLOGY

A. Definition of Centerline

Intuitionally, centerline is a curve that traverses the center of a hollow organ. In a simple shaped organ like colon, there is normally only one centerline, but as for some complicated shaped organs such as blood vessels, they will contain several centerlines that form a skeleton of the organ, and so called skeleton accordingly. In this paper, since we will only focus on the colon, we still use the term of "centerline". A more concise definition of centerline (or skeleton) is that the locus of centers of maximal balls contained in the shape [4]. Unfortunately, that definition is hard to realize because of its complexity, but the definition does give a good description of the centerline. It gives 3 important qualities that a centerline should have: Connectivity, Centricity and Singularity [5]. They are also used as an evaluating criterion of a centerline.

B. Overview of Centerline Algorithms

Since the proposition of path planning, many centerline algorithms were invented. In general, these algorithms can be divided in three classes: manually marking, topological thinning and distance mapping.

1) *Manually marking:* This method requires user to mark some key points through the center of the colon, and then it'll compute out the curve by the interpolation. This methods is simple in technology, but complicated even tedious in operation. Because that the colon is usually visualized on 2D screen, it is hard for user to locate the center point of the colon. Furthermore, this method cannot guarantee the centricity due to human mistakes.

2) *Topological thinning*: Topological-thinning algorithm peels off the colon voxels layer by layer, until the singularity is met while guaranteeing the connectivity. The idea of topological thinning is simple and it is traditionally considered to provide high quality centerline. The problem is that this algorithm is computationally expensive and many researches [6] have been addressed their efforts to speed up the peeling process.

3) *Distance mapping*: Distance mapping algorithm is the most fast one among the three algorithms. It treats the volume as a graph, and every voxel is a node when an edge is added between every two connected voxels. In this class of algorithms, the centerline can be defined as the Dijkstra

0-7803-8740-6/05/$20.00 ©2005 IEEE.

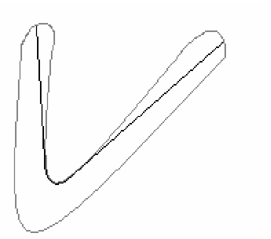

Fig.1. The centerline hugs the corner at a sharp turn

shortest path from a start voxel to an end voxel. Although, distance mapping can meet the connectivity and singularity well, its centricity quality is not high, but tents to hug the corner (Fig.1), especially at the sharp turn. Many researches were devoted on this problem. Ingmar Bitter *et al.* brought in a kind of penalty distance to the boundary nodes to make the centerline leaving away from boundary [7]. The algorithm proposed by Ming Wang *et al.* firstly computed the distance from boundary (DFB) for every node, then weighted the edge connect to node p with 1/*DFB*(p) [5]. It solves the hugging problem simply and effectively.

C. Description of the proposed Algorithm

Our proposed algorithm is based on distance mapping and is designed for VC systems. It consists of four components named colon segmentation, 3D Euclidean Distance Transform (EDT), boundary voxels cutting (BVC) and Dijkstra extraction (DE). It was proved that the new algorithm not only provides a high quality centerline, but also speeds up the former algorithms significantly.

1) *Colon segmentation*: The raw CT or MRI data contains all organs including colon, and both the topological thinning and distance mapping algorithm won't work when other organs exist in the data. So that segmenting the colon from the raw data is indispensable. There are many different algorithms proposed for segmentation such as level set [8], fast marching [9], morphological reconstruction [10] and watershed [11]. The traditional medical image segmentation is a time consuming process. For example, it takes more than 15 minutes for morphological reconstruction to segment the colon from a volume data (300*250*548 in size) on a PC.

Due to the simple shape and low intensity properties of colon, we employ a simple and fast segmentation algorithm for colon data. The segmentation algorithm takes two steps: threshold and branch detecting.

Threshold operation cut off any voxel whose intensity is out of the range of colon voxels. Then, the remaining voxels were divided into several unconnected brunches that are detected and merged together as the colon regions.

The threshold and branch detecting process is much faster than traditional segmentation algorithms. Experiment shows that this segmentation algorithm takes only 1 minute to segment the colon, which is 15 times faster than the morphological reconstruction.

Fig 2. shows the segmentation process of a colon data.

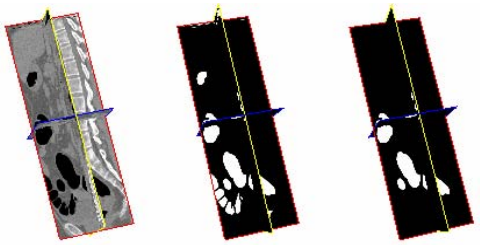

Fig.2. Segmentation. Left: original CT volume; Middle: volume after threshold; Right: volume after branch detecting

2) *3D Euclidean Distance Transform* (EDT): Centerline extracted by Dijkstra shortest path algorithm [12] tends to hug the corner at sharp turn (Fig.1) simply because hugging leads to a shorter path. At the hugging corner the centerline will lose its centricity and will misplace the virtual endo-camera. The hugging problem is due to equal treatment of boundary voxels and center voxels. 3D Euclidean Distance Transform calculates the Euclidean distance from every voxel to the colon boundary (DFB) [13], thereby it differentiates these voxels quantitatively. Furthermore, DFB is also the precondition of the boundary voxel cutting described in next section.

In this paper, we introduce a new simple and fast (in linear time) EDT algorithm, named *Bidirectional Distance Adjustion* (BDA) algorithm. It uses 10-14-17 metrix to approximate the Euclidean metrix($1-\sqrt{2}-\sqrt{3}$)in order to speed up the computation.

In BDA, The segmented volume is defined by following equation:

$$E(n0,n1,n2)=\{0,1\}, n0\in [0,N0],\ n1\in [0,N1],\ n2\in [0,N2]$$

Where, N0,N1 and N2 is the size in the 3 ortho-axis; *n0*, *n1* and *n2* are the coordinates position of the voxel; and E(*n0*,*n1*,*n2*)is the intensity of the voxel(0 means background voxel). Then we define D(*p*) as the distance from voxel *p* to the colon boundary in processing, P(*n0*,*n1*,*n2*)as the current voxel, Neib(*p*) as a set of voxels which connected to *p* and d(*p1*,*p2*) as the distance between two connected voxels *p1* and *p2* (10,14 or 17 in this paper). The adjusting for a single voxel *p*(*n0*,*n1*,*n2*) takes 3 steps as:

step 1) $D(p)=I(p)$

step 2) $\forall\ p' \in Nei(p)$, *if* $D(p)>D(p')+d(p,p')$, then $D(p)=D(p')+d(p,p')$, meanwhile, if $D(p')>D(p)+d(p,p')$, then $D(p')=D(p)+d(p,p')$.

step 3) $I(p)=D(p)$; $\forall\ p'\in Nei(p)$, $I(p')=D(p)$.

The single voxel adjusting can be separated into two kinds. Clockwise adjusting travels Nei(*p*) from low-left voxel to up-right one, while anticlockwise adjusting travels Nei(*p*) in the opposite order. With that, the BDA can be described as follow:

(1) $\forall p$, $I(p)= +\infty$.

(2) while *i,j,k* increase from 0,0,0 to *N0,N1,N2,* do the clockwise adjusting for *P(i,j,k)*.

(3) while *i,j,k* decrease from *N0,N1,N2* to 0,0,0, do the anticlockwise adjusting for *P(i,j,k)*

At the end, the result I(p) equals to DBF(*p*).

3) *Boundary voxels cutting* (BVC): The Dijkstra shortest path calculation is very time consuming because the colon volume usually has huge voxels, and we find that most colon voxels are nearby colon boundary while contribute nothing to the centerline. So that cutting off the boundary voxels does not affect the centricity of the resulted centerline, but speeds up Dijkstra shortest path significantly.

As we know, the DFB derived from EDT process provides good information to distinguish boundary voxels from center voxels. But unfortunately, the DFB is local information, and it is hard to peel off boundary as much as possible while guaranteeing the connectivity of the remaining voxels. In this paper, we define Maximum Center Voxel (MCV) as the voxels that have the largest DFB among its neighbor voxels. The MCV have good centricity quality, but due to locality, bend and ruga will bring on many MCV not belong to the centerline, so it is hard to generate the centerline from the center MCV by interpolation. Here, we propose an expanding method to firstly expand the MCV simultaneously until a connected voxels set is achieved, and then cut off all others not in the set. We call the operation as simultaneous branch expanding (SBE) in this paper. It is proved that the SBE process is faster and more effective. Averagely, this process will cut off 2/3 of the original colon voxels. One example result of BVC is shown in Fig.3.

The SBE treats the MCV as unconnected branches initially then expands them one by one until all the branches connect to each other. We define B as the unconnected branch set which is the very MCV like at the beginning, cn as the count of branches, and mn as the number of branches merged together. The algorithm stops when mn equals cn-1 and then all the branches will be merged together. Fig.4 shows the flowchart of the SBE algorithm.

4) *Dijkstra extraction* (DE): To avoid the hugging problem and improve the centricity, Ming Wang weighted the edges which connect to voxel p as 1/DBF(p) [5]. This method is simple and costless, but 1/DBF(p) does not decrease equably as DBF(p) increases. The 1/DBF(p) decreases faster near the boundary but slower near the center. Contrarily, Our proposed algorithm weights the edge with *MAX*-DBF(p), where *MAX* is a big enough integer guaranteeing *MAX*-DBF(p) nonnegative.

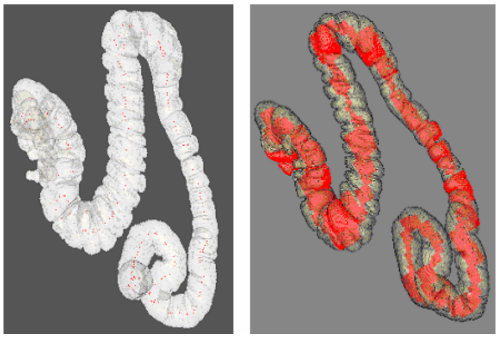

Fig.3. BVC Process. Left: colon and its MCV. Right: BVC cuts off most of the colon voxels, only about 1/3 of the total voxels are left over around the center of the colon.

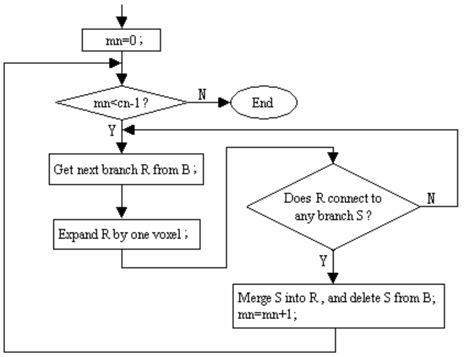

Fig.4. Flowchart of the simultaneous branch expanding (SBE)

In the proposed algorithm, user can specify the start voxel for Dijkstra algorithm. If user does not define it, the algorithm will choose any voxel to start and run the Dijkstra algorithm. Then it chose the voxel that is the farthest away from that voxel as the start voxels when the approach runs the Dijkstra algorithm and chose the farthest voxel away from start voxel as the end voxel. After that, the centerline will be generated as the shortest path between the start voxel and the end voxel. Fig.5 shows an extracted centerline example using the proposed algorithm.

III. EXPERIMENTAL RESULTS

We tested the new algorithm by two CT datasets. Both datasets were generated by a Siemens CT scanner. The size of data 1 is 512*512*549 with pixel span of 0.71mm, and the slice span is 1.0 mm when data 2 is 512*512*320, 0.8mm, and 1.5mm respectively. The proposed approach is programmed and run on a 2.66G Hz, 1.0G memory PC platform.

To measure the efficiency of BVC, the first experiment extracts centerline without BVC, while the second experiment does the BVC calculation before DE. Table I and table II conclude the results.

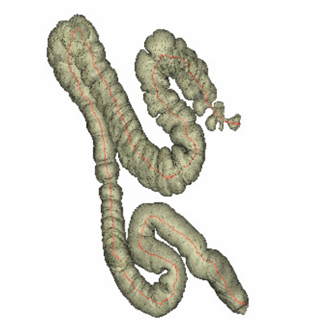

Fig.5. An extracted centerline example using the proposed algorithm

Table I

Experimental Results without the Boundary Voxels Cutting

Data	**Size**			**BDA (sec)**	**DE (sec)**	**Total (sec)**
	X	Y	Z			
1	512	512	549	4	25	29
2	512	512	320	3	17	20
Average	--	--	--	3.5	21	24.5

Table II

Experimental Results with the Boundary Voxels Cutting

Data	Colon voxels		BDA (sec)	BVC (sec)	DE (sec)	Total (sec)
	Before Cutting	After Cutting				
1	2305796	810687	4	2	5	11
2	1436972	400892	3	1	2	6
Average	1871384	605789	3.5	1.5	3.5	8.5

Table I and table II show the high efficiency of the boundary voxels cutting. It deleted 2/3 of the original voxels, and saved 17.5 seconds for Dijkstra algorithm with the expense of only 1.5 second averagely. As the result, it speeded up the original algorithm by nearly 2 times. Meanwhile, the boundary voxels cutting does not affect the centerline result at all.

In our VC system designed for clinic application, the new centerline algorithm was employed to serve as the base of its navigation model. With the extracted centerline, we realized an automatic navigation method, which can guide the virtual endo-camera fly through the inner colon. Meanwhile user can stop the flight anywhere and control the virtual endo-camera manually. Whenever finished the operation, he can resume the auto-flight from where it was stopped. Fig.6 shows an user interface of the VC system.

IV. CONCLUSION

In this paper, we proposed a novel centerline extraction algorithm based on distance mapping algorithm for a VC application. A fast and feasible segmentation method is employed to segment define the colon regions. Furthermore, we induct an efficient boundary voxels cutting algorithm based on the DBF information, which greatly speeds up the Dijkstra centerline extraction process. The experimental results prove that the new algorithm improves the distance mapping algorithms significantly and the new centerline algorithm is 3 times faster than the existing distance mapping algorithms.

In the future, we will improve this algorithm, and strengthen it to extract skeletons for more complicated shape organs such as blood vessels and airways in the lungs.

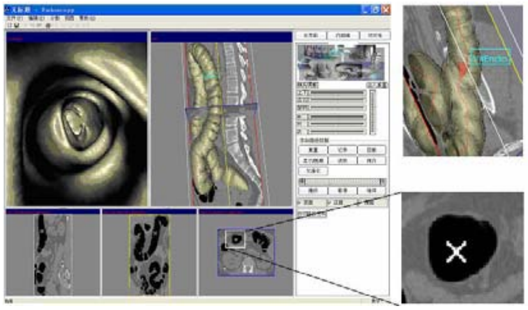

Fig.6. The VC system in which we use the new centerline algorithm for path planning

V. ACKNOWLEDGEMENT

Thank to the Shanghai 6th people hospital, which provides the test colon data, and also to Jianfeng Xu for his great advice to the work. This work is partially supported by NDI and the Shanghai municipal research fund.

.

REFRENCE

[1] Delphine Nain, Steven Haker, Ron Kikinis, W.Eric L.Grimson, “An Interactive Virtual Endoscopy Tool,” Satellite Workshop at the Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2001), 2001.

[2] Rainer Wegenkittl, Anna Vilanova, Balint Hegedus, “Mastering Interactive Virtual Bronchioscopy on a Low-End PC,” Visualization 2000. Proceedings , pp.461 - 464, 590

[3] Anna Vilanova, Andreas König, Eduard Gröller, “VirEn:A Virtual Endoscopy,” System. Machine GRAPHICS & VISION, vol.8(3), 1999, pp.469-487

[4] H. Blum, A transformation for extracting new parameter of shape. Models for the Perception of Speech and Visual form, Cambridge,MA: MIT Press,

[5] Ming Wan, Zhengrong Liang, Qi ke, Lichan Hong, Ingmar Bitter, Arie Kaufman, “Automatic Centerline Extraction for Virtual Colonoscopy,” IEEE Transactions On Medical Imaging, vol.21(12), 2002, pp.1450-1460

[7] D.S. Paik, C.F. Beaulieu, R.B. Jeffery, G.D. Rubin, and S. Napel, “Automated Flight Path Planning for Virtual Endoscopy,” Medical Physics, vol.25(5), 1998, pp.629-637

[6] Ingmar Bitter, Arie E. Kaufman, Mie Sato, “Penalized-Distance Volumetric Skeleton Algorithm,” IEEE Transactions On Visualization And Computer Graphics, vol.7(3), 2001, pp.195-206

[8] Osher, S., and Sethian, J.A, “Fronts Propagating with Curvature Dependent Speed: Algo-rithms Based on Hamilton-Jacobi Formulation,” J. of Comput. Phys., vol. 79 ,1988, pp. 12-49

[9] Sethian, J.A, “Fast marching methods,” SIAM Review, vol.41(2), 1999, pp.199~235.

[10] L.Gu, T.Kaneko, “Extraction of Organs Using Three-Dimensional Mathematical Morphology”, Systems and Computers in Japan, Vol.31-7, 2000, pp.29-37.

[11] S. Beucher and F. Meyer, “The morphological approach to segmentation:The watershed transform,” in Mathematical Morphology in Image Processing, E. R. Dougherty, Ed. New York: Marcel Dekker, 1967

[12] E.Dijkstra, “A note on two problems in Connexion the Graphs,” Numer. Math., vol.1(1),1959, pp.269-271

[13] Frank Y. Shih, Yi-Ta Wu, “The Efficient Algorithm for Achieving Euclidean Distance Transformation,” IEEE Transactions On Image Processing, vol.13(8), 2004, pp.1078-1091