Fast and Robust Extraction of Centerlines in 3D Tubular Structures Using a Scattered-Snakelet Approach

Christoph Spuhler, Matthias Harders, Gábor Székely

Computer Vision Laboratory, Swiss Federal Institute of Technology, CH-8092 Zurich, Switzerland

ABSTRACT

We present a fast and robust approach for automatic centerline extraction of tubular structures. The underlying idea is to cut traditional snakes into a set of shorter, independent segments - so-called snakelets. Following the same variational principles, each snakelet acts locally and extracts a subpart of the overall structure. After a parallel optimization step, outliers are detected and the remaining segments then form an implicit centerline. No manual initialization of the snakelets is necessary, which represents one advantage of the method. Moreover, computational complexity does not directly depend on dataset size, but on the number of snake segments necessary to cover the structure of interest, resulting in short computation times. Lastly, the approach is robust even for very complex datasets such as the small intestine. Our approach was tested on several medical datasets (CT datasets of colon, small bowel, and blood vessels) and yielded smooth, connected centerlines with few or no branches. The computation time needed is less than a minute using standard computing hardware.

Keywords: Centerline Extraction, Skeletonization, Snakes, Virtual Endoscopy, Virtual Colonoscopy

1. INTRODUCTION

The motivation for our research is the support of diagnosis and treatment for the small intestine. In previous work,[1] we have presented an interactive segmentation tool using haptic feedback to support the extraction of the organ's centerline. On the one hand, a centerline of the small intestine is necessary to allow distance measurements along its length, facilitating the localization of pathologies with respect to the different organ segments. On the other hand, the centerline can be used to perform a virtual endoscopy of the organ. Virtual endoscopy of the colon, or virtual colonoscopy, has already become an accepted alternative to conventional colonoscopy.[2] It's advantage lies in the increased comfort for the patient, as well as shorter examination times and avoiding possible perforation of the colon with the endoscope. Clinical tests with virtual endoscopy of the small intestine[3] have shown that the method is reliable for detecting lesions. The procedure has not become common, however, due to the difficulty of finding a central path through the small intestine. Methods developed for finding the centerline of the colon fail on the small intestine due to the much higher complexity of the organ. Yet, virtual endoscopy of the small intestine is very much needed, since the organ is not fully accessible by a conventional endoscope. Alternative methods using a camera capsule[4] allow only an incomplete visualization of the small bowel, and are unable to provide sufficiently precise anatomical localization for the acquired images.

In this paper we present a novel scattered snakelet approach, which automatically segments the majority of the small intestine, leaving only a few sections to be completed by the user. The extracted partial centerline allows us to increase the speed and accuracy of the segmentation in our previously developed haptically assisted segmentation system. The completed segments are used to provide explicit haptic guidance along the length of the tube. Those segments of the tube which could not be automatically segmented, can be completed using the original method. Most previously proposed algorithms are unsuitable for this partial centerline extraction because they rely on the assumption that a complete centerline can be extracted. Moreover, in the proposed method, computational complexity is reduced and depends mainly on the number of initialized snakelets, not the dataset size. Finally, on simple datasets such as the colon, the method gives similar results to other methods;

Further author information: (Send correspondence to Christoph Spuhler)
Christoph Spuhler: E-mail: spuhler@vision.ee.ethz.ch, Telephone: +41 44 632 4937

a complete and smooth centerline without human interaction and with very little computation. A standard clinical dataset for virtual colonoscopy can be processed in less than 30 seconds. Computation times of under one minute are achieved for the small intestine.

2. PREVIOUS WORK

A number of methods for the extraction of the colon centerline have been described in literature. However, up to now, no approaches have been presented for the small intestine due to problems related to its complex geometry and tight packing. We therefore review some of the methods which are commonly used for the automatic extraction of centerlines for the colon.

Most methods presented recently have been based on some kind of topological thinning[5,6,7,8,9] . Topological thinning is analogous to peeling an onion; voxels are removed from the object until only a voxel-wide centerline remains. Usually the method is divided into two steps. First the object is thinned, leaving a rough centerline. This centerline is then traversed in order to remove extranous loops and branches and to fill any gaps, so that a singular and complete line remains from the beginning to the end. Topological thinning has traditional suffered from long computation times. However, recent optimized approaches have brought the calculation times from over 10 minutes down to under a minute[7] for colon datasets.

Topological thinning runs into severe limitations when applied to the more difficult case of the small intestine. Topological thinning by definition preserves the topology of the given object. However, in the small intestine we run into the problem that different parts of the organ are touching. Due to degraded image quality it is not uncommon for sections of the wall to be missing in the image or to be very thin. In addition, in the image the tube can have gaps, due to missing contrast agent or tightening and folding of the intestine. Topological thinning also suffers from branching. Previous algorithms have used the knowledge of a starting and endpoint to remove these branches. In the small intestine this is not possible because we cannot assume that the extracted line is complete from beginning to end. It is the combination of these problems that makes the use of topological thinning unsuitable for the small intestine.

Deschamps *et al*[10] use Front-Propagation to find central lines in medical structures. The Front-Propagation uses Fast-Marching and Level-Set methods to propagate a front of voxels along tubular objects. The speed at each voxel is controlled by a Euclidean distance measure calculated from the wall of the object. Because the front cannot pass gaps or very tight spots, it would have to be initialized several times in order to partially segment objects such as the small intestine.

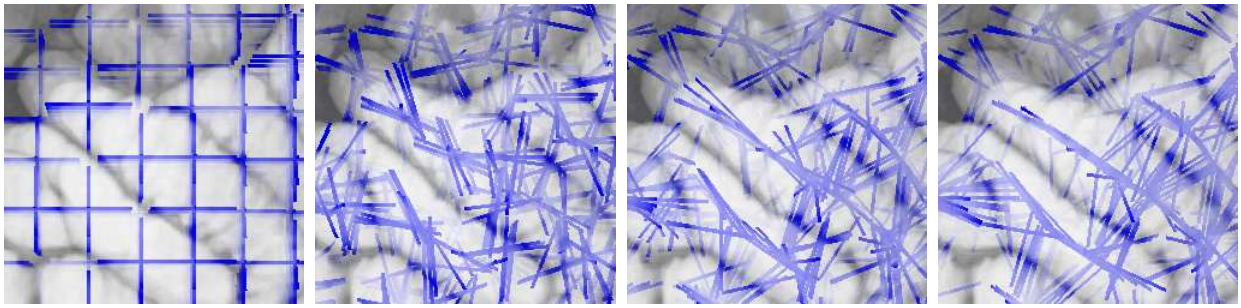
Chaudhuri *et al*[11] use a graph search to segment a colon. They iteratively subdivide the volume inside the colon into blocks until all the blocks are contained entirely within the walls of the colon. They then use Dijkstra's algorithm to find an optimal path connecting the blocks. In the small intestine the wall can ususally not be completely identified, due to the tight packing of its loops. In addition, gaps in the tube would prevent the blocks from being connected.

3. METHOD

The proposed method is meant to be used for difficult medical structures which cannot be segmented by any of the methods above, such as the small intestine. For this reason, we expect that the object to be segmented has large gaps and an unclear and noisy boundary. In addition, we take into account that the object might not be entirely segmentable without human intervention. The aim is to automatically segment as much as possible of an object, producing smooth, non-branching and correct centerline segments. In this manner, most of the dataset can be segmented automatically, leaving only a few spots to be examined with user input.

Since the segmentation of the entire path might not be possible, we divide the problem into small intervals, a kind of "divide and conquer approach". Furthermore, we make the following assumptions:

- Using the a Euclidean distance map of the volume, we can find single points at more or less regular intervals which lie at the center of the small intestine.



(a) (b) (c) (d)

Figure 1. (a) Initialization of a regular grid of snakelets covering the volume. (b) to (d) The snakelet endpoints are iteratively refined to the center of the small-intestine. The various outliers will be removed in a later step.

- Between two points lying at short distances from each other in the center of the small intestine, it is possible to find an optimal centered path in the tube between them.
- We can detect whether a given path segment runs along the center of the tube for its entire length or crosses a wall and runs into another section of the tube.

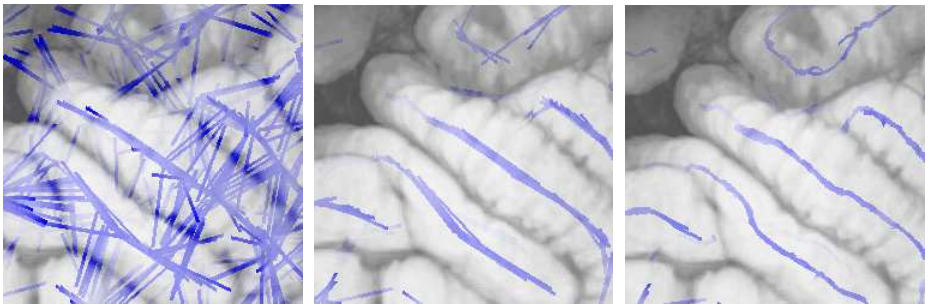
For these assumptions to be true some criteria need to be met. Firstly, all voxels inside the tube should be reliably classified as belonging to the tube, since otherwise a Euclidean distance map cannot be correctly calculated. Secondly, the individual segments should not be too long so that the calculation of an optimal path is possible without cutting corners. Thirdly, if a wall between parts of the tube is too thin to be detected, wrong segments cannot be ruled out.

We define a centerline as a line following a tubular object at the maximum distance away from its boundaries. It therefore follows that the distance map has a ridge of maximal distance values running along the center of the tubular object. The values on this ridge vary depending on the thickness of the tube at a particular point, however, the gradient in the distancemap always points towards this ridge. From the centerline, various minor ridges might branch off, due to noise on the surface of the tube. However, the gradient along these in all but the most extreme cases again points towards the main ridge at the center of the tube. When locating the endpoints of our snakelets, we search for a local maximum in the distance map by following the distance map gradient, in order to find points on this main central ridge.

Keeping these criteria in mind, we calculate centerline segments for all parts of the volume object. We will call these segments 'snakelets'. The calculation of snakelets is done in four steps. First, straight snakelets are initialized (or scattered) in a grid-like fashion so as to cover the entire dataset. Next, the endpoints of each snakelet are iteratively moved to the nearest local maximum in the distance map. Then, the remaining points of the snakelet are adapted to follow the center of the distance map between the two endpoints using the active contour method. Lastly, snakelets crossing the object wall into other sections of the tube are removed using a wall detection. For the refinement of the endpoints, the active contour model and the edge crossing we use subpixel accuracy as shown in Figure 3. This gives smoother results, better centerline fitting and more reliable detection of edges. The individual steps are described in more detail in the following sections.

3.1. Calculation of Euclidean distance map

The calculation of the snakelets is based on a Euclidean distance map of the object to be segmented. This distance map contains the distance at every point in the volume to the border of the tube. Inside the tube, the distance values are positive, outside they are negative. In this way the gradient of the distance map always points towards the central ridge, and therefore the center of the tube object. In order to calculate the distance map for the small intestine dataset, a binary threshold is applied to the initial radiological data. This yields a good approximation of the investigated lumen due to the application of contrast agent before the CT scan. In CT images, bone has nearly the same intensity as the contrast agent, and will be classified as as belonging to



(a) (b) (c)

Figure 2. (a) Shows the snakelets with centered endpoints. (b) The remaining snakelets after wall crossings are removed. (c) The snakelets after the points have been refined using the active contour model

the object as well. However, ample spatial distance between bone and the intestine allows us to ignore any bone which is segmented along with the intestine. Thereafter, Euclidean distances to the boundary are determined for all voxels using the algorithm described by Danielsson.[12] This algorithm scans the voxels in the datasets along all three axes in the forward and backward direction, and requires around 5 seconds for a 256x256x256 dataset.

3.2. Snakelet initialization

Snakelets are initialized on a grid at regular intervals in the volume; three at each position directed along the major axes. The intervals between adjacent snakelets in the grid are smaller than the snakelet length, thus causing the snakelets to overlap. This prevents gaps from appearing in the final centerline. The length of each snakelet should be adapted to the desired feature size. Longer snakelets generally result in a smoother central line while snakelets which are too short might cause some unwanted branching in the result. In general the snakelets length should be slightly larger than the diameter of the tubular structures. Figure 1a shows a grid of snakelets initialized for the small intestine.

The calculation time of the centerline is directly proportional to the number of snakelets which are initialized. To shorten the calculation we can initialize the snakelets only in parts of the volume where the object is present. Snakelets whose center corresponds to a distance map value smaller than zero are not initialized.

3.3. Endpoint refinement

At this point the snakelet is a straight segment defined by its endpoints. The endpoints of each snakelet are iteratively refined towards the center of the tube using the gradient of the distance map as a driving force. The endpoint position at each iteration is defined by

$$\vec{p}_{t+1} = \vec{p}_t + d \cdot G(\vec{p}_t) \tag{1}$$

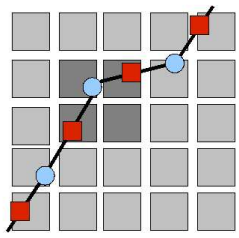


Figure 3. Endpoint refinement, active contour and edge detection are performed with subpixel accuracy. The 8 adjacent (four shown in 2D) voxels are trilinearly interpolated (shown in dark for the center point). The circles are the points used for the active contour. The squares are the additional sample points used for the edge detection.

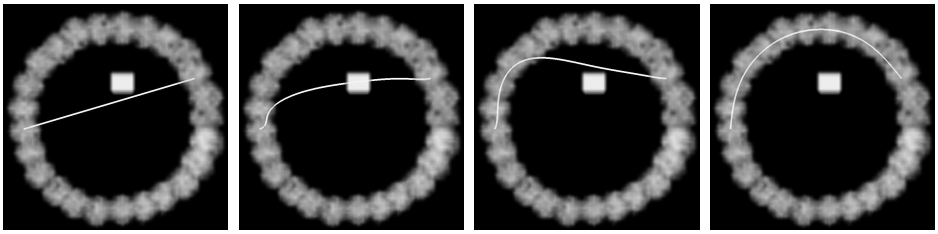


Figure 4. Iterative active contour refinement of a snakelet to lie in the center of a tubular structure

where G is the distance gradient, t is the time step and d is a scaling factor. At each iteration only one endpoint is moved to avoid "wandering" of the snakelet. Also, after each iteration the snakelet is scaled to its initial length to avoid stretching or shrinking. Iteration is continued until the position stabilizes. It can then be assumed that the endpoint is located on the main ridge of the distance map and therefore in the center of the tube. Figure 1 shows the iterative endpoint refinement for the small intestine.

3.4. Snake refinement

From the previous step, the endpoints of each snakelet now lie at the center ridge of the distance map. Between the endpoints they are connected by a straight line segment. Next, the remaining points of the snakelet are adjusted to follow the center of the tubular structure using an active contour model, similar to the one described by Neuenschwander *et al.*[13] The model again uses the distance map gradient as the external force to adapt to the centerline of tubular structures. The position of its points is given by the differential equation

$$-\alpha v_{ss} + \beta v_{ssss} = -P_v \tag{2}$$

where v is the position, and P is the position dependent force acting on a point. The parameters α and β determine the smoothness of the resulting snakelet, and therefore the final centerline and have been chosen empirically. For the colon, we use a snakelet with 15 points (roughly equal to the pixel length), a tension parameter (α) of 0.005 and rigidity (β) of 0.00005. The processing time for this step is on the order of 10 seconds for the colon. Figure 4 shows a single snakelet whose path is refined to optimally follow the center of a tubular object. Due to the ziplocked nature of the active contour, the snakelet does not get caught on the block placed in the middle of the spiral.

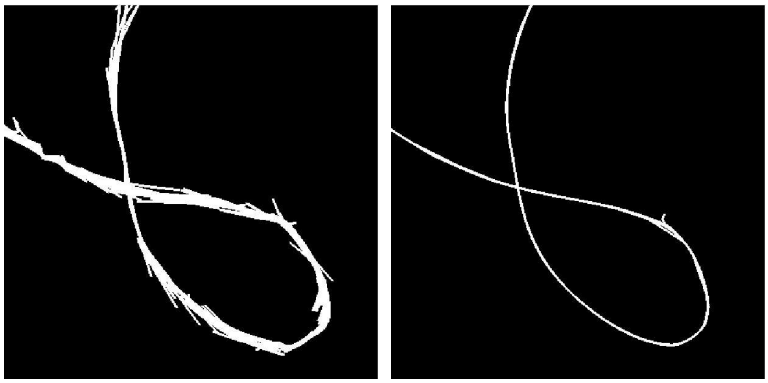
3.5. Cropping

At this point we have a multitude of snakelets that lie at the center of various tubular structures in the image. However, we have not yet accounted for snakelets that cross from one segment of the tube into an adjacent one, crossing a wall in the process. The endpoints of these snakelets both lie in the center of the tube, yet they are not part of the centerline and need to be removed. In order to find and eliminate these outliers we detect whether a snakelet crosses an edge at any point along its path. For this we examine the voxels along the path of the snakelet. A voxel value below the threshold of the contrast intensity is considered as being part of the wall. If any such voxels exist along the path of the snakelet, the snakelet is removed. As in the endpoint refinement and the active contour steps, we use subpixel accuracy to determine the voxel values along the snakelet. The sampling rate along the snakelet should be equal or higher than the voxel frequency, since otherwise we might miss an edge located between samples. We therefore insert additional sampling points between the points used for the active contour. Figure 2 shows the snakelets in the small intestine before and after cropping.

3.6. Snakelet merging

In some cases it is useful to combine the overlapping path segments which together constitute an implicit centerline into a single explicit path segment. For this we traverse the points of each snakelet and average it with adjacent snake segments. For a given point $\vec{p_0}$ on the snakelet we can define an orthogonal plane such that

$$\vec{d} \cdot (\vec{p} - \vec{p_0}) = 0 \tag{3}$$



(a) (b)

Figure 5. Mergin of adjacent snakelets to get a single centerline. (a) shows the implicit centerline formed by the snakelets following the center. Due to some noise in the distance map they are not exactly aligned (b) shows the snakelets after they are averaged together

where $\vec{d}$ is the tangent along the snakelet at $\vec{p_0}$ and the normal to the given plane. We then calculate the intersection of this plane with nearby snakelets. Only intersection points within a certain maximum distance are considered. The resulting intersection points are averaged together with $\vec{p_0}$ and $\vec{p_0}$ is moved to this new position. By applying the method to all points in every snakelet, all adjacent snakelets are brought to lie at the same averaged position. The implicit path of snakelets following the center of the tubular object can then easily be combined into an explicit centerline. For the snakelet averaging it is important to set a maximum distance for adjacent snakelets which are considered to be averaged together. This distance should be relatively small on the order of a few pixels, since otherwise we run the risk of averaging our path with other sections of the path running close by. For smooth and clearly separated datasets such as the colon, the distance can be made much larger; approximately the width of the tube. Figure 5 shows the merging algorithm applied to a section of the colon centerline.

4. RESULTS

The proposed algorithm has been applied to various medical datasets. Figure 6 shows the steps of the centerline extraction for a typical colon dataset (128x128x128 voxels). Processing time was under 30 seconds on a 2GHZ Pentium with 1 GB memory. Most of the processing time was used for the active contour refinement which was performed with subpixel accuracy. The resulting centerline is smooth and connected. Towards the end of the colon some unwanted branching is present due to the very wide organ geometry in this region. Figure 7 shows the result of the algorithm being applied to a small intestine dataset (256x256x256 voxels). Despite the difficult geometry of the organ, a correct centerline was extracted for most parts of the data. Sections of the centerline are missing in regions where the shape is unclear.

5. DISCUSSION

The presented algorithm can be used in two forms. It can be applied to simpler datasets in order to perform an entirely automatic segmentation, much as previous algorithms have done with the path finding for the colon. Secondly, the algorithm can be used as a first step in order to aid a human operator to more easily find a complete central path for complicated tubular objects, such as the small intestine. We believe that in the near future, the ability of a human operator to deduce a correct path in medical data through prior knowledge and visual inspection of the dataset as a whole, will not be matched by an automatic algorithm. However, the human interaction is often slow and tedious and can be greatly facilitated by computer guidance such that manual decisions about the path are limited to situations where they are truly needed. We have found that integrating the pre-extracted partial paths with our haptic segmentation system speeds up and facilitates the user interaction. In this segmentation system, the user moves along the intestine using a haptically guided cursor. In a previous

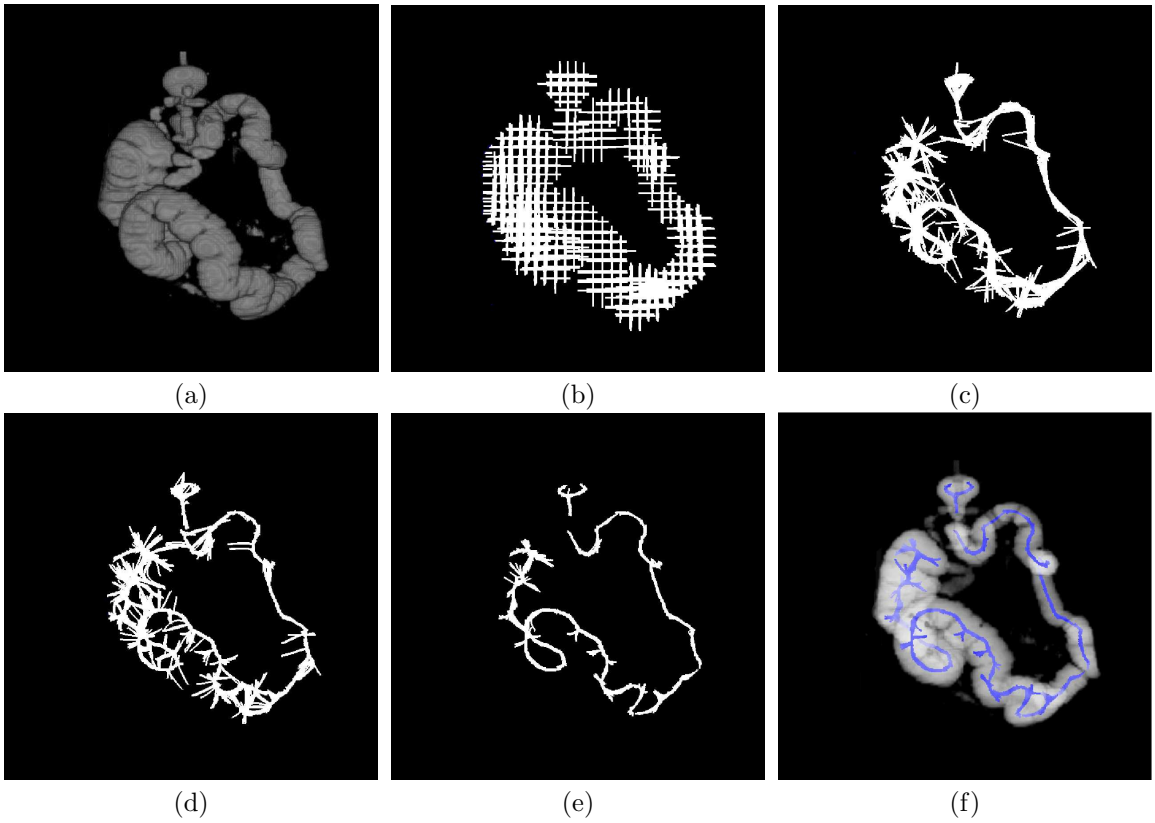


Figure 6. Finding central line through the colon dataset. Shown are: (a) the original dataset, (b) after snake initialization, (c) after endpoint refinement, (d) after active contour refinement, (e) and (f) the final centerline after cropping.

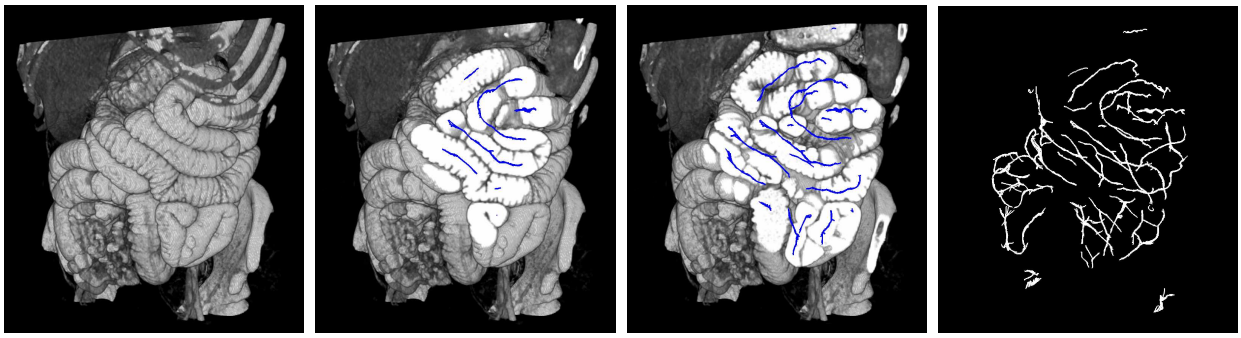


Figure 7. Extracting a partial central line through the small intestine dataset. (a) the original dataset. (b,c,d) cut-off views of varying degrees showing the extracted line following the tube. Segments along the tube are missing where the structure of the intestine is unclear.

version, the haptic feedback was generated solely on the basis of a distance-map, holding the cursor to the ridge of the maximum distance in the tube. This ridge, however, is often noisy, leading to perceptually disturbing or even misleading feedback to the user. Generating a force to hold the cursor to an explicit centerline whenever possible therefore results in a much smoother and better guidance for the user. This in turn results in better and faster segmentation times and overall better user acceptance.

6. CONCLUSION

We have developed an algorithm which uses scattered snakelets for finding the centerline of tubular structures. This algorithm is suitable for organs with high geometric complexity where a complete centerline is otherwise not possible. With the proposed algorithm, an almost complete centerline can be calculated for the small intestine. With a previously presented haptically assisted segmentation system, the missing segments can be quickly completed by a human operator. The partial centerline extraction facilitates and speeds up the finding of a complete centerline. The proposed algorithm is also suitable for simpler datasets such as the human colon. The algorithm is very fast, accurate, and yields connected and smooth centerlines without human interaction for such datasets.

REFERENCES

1. M. Harders and G. Szekely, "Improving medical segmentation with haptic interaction," *Proc. of IEEE Computer Society Conference on Virtual Reality* , pp. 243–250, 2002.
2. D. Vining, "Virtual endoscopy: is it reality?," *Radiology* **200**, pp. 30–31, 1996.
3. P. Rogalla, M. Werner-Rustner, A. Huitema, A. van Est, N. Meiri, and B. Hamm, "Virtual endoscopy of the small bowel: phantom study and preliminary clinical results," *European radiology* **8**, pp. 563–567, 1998.
4. P. Kongkam and et al, "Result of wireless capsule endoscopy in patients with suspected small bowel disease: The first series from thailand," *Journal of the Medical Association of Thailand* **97**, pp. 35–39, 2004.
5. R. Chiou, L. Zhengrong, L. Hong, and M. Achniotou, "An interactive fly-path planning using potential fields and cell decomposition for virtual endoscopy," 1999.
6. I. Bitter, A. Kaufman, and M. Sato, "Penalized-distance volumetric skeleton algorithm," *IEEE Transactions on Visualization and Computer Graphics* **7**(3), 2001.
7. R. Sadleir and P. Whelan, "Colon centerline calculation for ct colonography using optimised 3d topological thinning," *3DPVT* , pp. 800–803, 2002.
8. S. Bouix, K. Siddiqi, and A. Tannenbaum, "Flux driven fly throughs," *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition* , pp. 449–454, 2003.
9. A. Telea and A. Vilanova, "A robust level-set algorithm for centerline extraction," in *VISSYM '03: Proceedings of the Symposium on Data Visualisation*, pp. 185–194, Eurographics Association, (Aire-la-Ville, Switzerland, Switzerland), 2003.
10. T. Deschamps and L. Cohen, "Fast extraction of minimal paths in 3D images and applications to virtual endoscopy," *Medical Image Analysis* **5**, pp. 281–299, Dec. 2001.
11. P. Chaudhuri, R. Khandekar, D. Sethi, and P. Kalra, "An efficient central path algorithm for virtual navigation," *Proceeding of the Computer Graphics International* , pp. 188–195, 2004.
12. P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and Image Processing* **14**, pp. 227–248 (1980).
13. W. M. Neuenschwander, P. Fua, L. Iverson, G. Szkely, and O. Kbler, "Ziplock snakes," *International Journal of Computer Vision* **25**, pp. 191–201, 12 1997.