

Clinical evaluation of an automatic path tracker for virtual colonoscopy

Roel Truyen¹, Thomas Deschamps², and Laurent D. Cohen³

¹ Philips Medical Systems Nederland,
EasyVision Advanced Development
`roel.truyen@philips.com`

² Laboratoire d'Electronique Philips France,
`thomas.deschamps@philips.com`

³ Laboratoire CEREMADE, Université Paris Dauphine, France,
`laurent.cohen@ceremade.dauphine.fr`

Abstract. Virtual colonoscopy is a minimally invasive technique allowing early detection of colorectal polyps. A path or centerline through the colon can be very useful to perform virtual endoscopy. Manual path tracking is a very time-consuming task and the resulting path depends a lot on the experience of the operator. This severely limits the applicability of the path-based visualization and inspection methods.

An automatic path tracker for virtual endoscopy was introduced in [3], based on previous work on minimal path extraction ([1]). First, we briefly recall the theory of the automatic path tracker, detailing how we adapt this method for virtual colonoscopy. We show the speed and robustness of this automatic path tracker by means of a multi-user study where we measured the total user time and the difference in results between users on 29 clinical cases.

1 Introduction

The physician's efficacy in virtual colonoscopy can be increased by using an automatic path tracking tool. This tool must be easy to learn and operate, fast and the result should be largely operator-independent. In that case, a less experienced technologist can track a path during a preprocessing step, reducing the time spent by the physician. We have used an automatic path tracker that was first presented for virtual endoscopy in [2]. Based on the theory of minimal paths [1], several improvements were brought in [3] to accelerate the computations.

In this paper we briefly recall the theory of minimal paths, then we detail how we apply the method of [1] to the particular case of colonoscopy, and finally we present a clinical evaluation of the path tracker. In section 2, we describe the method to automatically extract a path between user defined start and end points. In section 3, we detail how we applied the technique for path tracking in the CT colon. The results of a clinical multi-user validation are presented in section 4. A short conclusion summarizes the benefits of the automatic path tracker.

2 Finding a path in a 3D tubular structure

2.1 Path of minimal cost

We formulate the path construction as a minimal cost path search. As an example, see Fig. 1 where we want to extract a path in a vessel between two points. If we use the grey level itself as the local cost, the minimal cost path will follow the dark lumen of the vessel as much as possible.

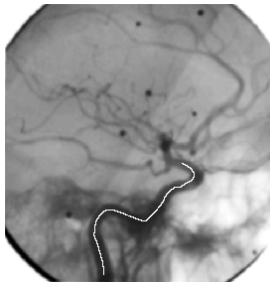


Fig. 1. Extracting a path on a digital subtracted angiography (DSA) image

In [1], the authors proposed to extract the minimal path by formulating a simplified snake energy model for the path C , without the second derivative term of the classical formulation of [5]:

$$E(C) = \int_{\Omega} \{w + P(C(s))\} ds = \int_{\Omega} \{\tilde{P}(C(s))\} ds . \quad (1)$$

where Ω is the domain of definition of C , s represents the arc-length parameter, and $w > 0$ is a regularization term which constrains the length and the curvature of C , as shown in [1]. The potential term P only depends on image grey-value information, and should be low at the image positions where the path should go.

First step is to compute the *minimal action map* $U(p)$, which is defined as the cost of the minimal cost path between a starting point p_0 and any point p :

$$U(p) = \inf_{p_0, p} \left\{ \int_{\Omega} \tilde{P}(C(s)) ds \right\} . \quad (2)$$

It can be shown that U satisfies the *Eikonal* equation, that is also used in geometric optics to describe lightwave propagation.

$$\|\nabla U\| = \tilde{P} . \quad (3)$$

The minimal action map U computed according to the Eikonal equation (3) has a global minimum $U(p_0) = 0$ at the start point p_0 .

In a second step, the minimal cost path is extracted by a simple gradient descent on the *minimal action map* $U(p)$ from the desired end point p_e back to the starting point p_0 .

2.2 Fast Marching algorithm

In order to discretize equation 3, authors in [3] based on previous work in [7] have proposed an iterative method to calculate the unknown u at each voxel (i, j, k) as:

$$\begin{aligned} & (\max\{u - U_{i-1,j,k}, u - U_{i+1,j,k}, 0\})^2 + \\ & (\max\{u - U_{i,j-1,k}, u - U_{i,j+1,k}, 0\})^2 + \\ & (\max\{u - U_{i,j,k-1}, u - U_{i,j,k+1}, 0\})^2 = \tilde{P}_{i,j,k}^2 . \end{aligned} \quad (4)$$

giving the correct viscosity-solution u for $U_{i,j,k}$. This quadratic equation implies that the action U at (i, j, k) depends only on the 6-neighbors that have lower action values, and that the action can only grow when iterating.

Based on this observation, the *fast marching* technique introduces order in the selection of the voxels, and the Eikonal equation can be solved using a front propagation technique. Starting from the initial point p_0 with $U = 0$, this algorithm chooses at each iteration the voxel with minimum action value and solves equation (4) in its 6-neighborhood. This corresponds to a wavefront that expands with a speed inversely proportional to the local potential \tilde{P} . Wavefront expansion will stop when the desired end point p_e is reached.

To perform these operations efficiently with $O(N \log_2 N)$ complexity on a N voxel grid, the voxels are stored in a min-heap data structure (see details in [7]). It is shown in [3], that when we have reached the end point p_e , we have visited the necessary set of voxels for the path construction. Therefore, we can constrain the computations to this set of voxels, thus saving computation time.

3 Application to virtual colonoscopy

3.1 Problem formulation

In virtual colonoscopy, we want to extract the colon centerline as a navigation guide for a virtual endoscopic camera. This centerline should stay in the colon lumen, pass through the center as much as possible, start and end at user defined points, and be smooth. User interaction can be reduced by automatically detecting the end point of the path.

3.2 Cost function

In a CT colon scan, the contrast between the air-filled colon lumen and the surrounding tissue is very high, and the Hounsfield values $I(\mathbf{x})$ of both structures are very constant. Therefore, we can simply use a thresholding function as the potential in each voxel: $P(\mathbf{x}) = 0$ if $I(\mathbf{x}) < T$ and $P(\mathbf{x}) = C$ elsewhere.

The minimal cost path using this thresholded cost will be the path with minimal Euclidean length between the start and end point that still stays inside the colon.

3.3 Path Centering

If we use the cost function defined in section 3.2, the minimal path will not be centered. The effect near a turn in the structure is shown in Fig. 2-left (dashed line) and a virtual view in Fig. 2-middle. This is due to the fact that the Euclidean path length is minimized, and no cost benefit is given for centering.

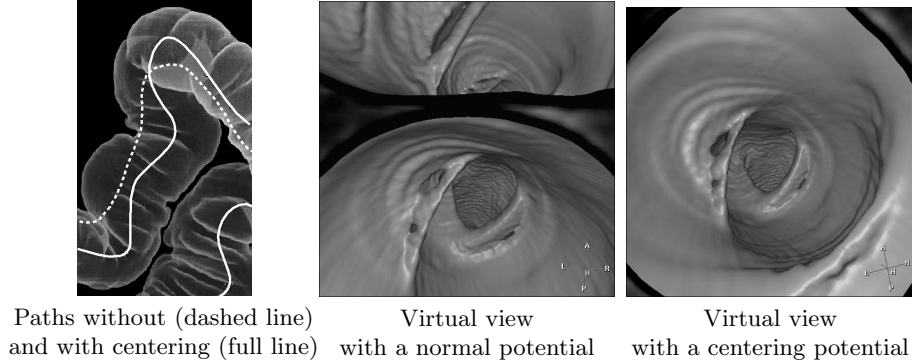


Fig. 2. Centering the path in the colon

In [3] a technique is detailed how to center paths in a tubular shape, using the front propagation method at each step. A short outline of the method:

1. The propagated front is used to obtain a segmentation of the colon, in a way similar to [6].
2. The distance \mathcal{D} to the colon wall is calculated in every voxel using a front propagation with constant potential. This distance is used to define a new potential P' as $P'(\mathbf{x}) = A \exp(-B\mathcal{D})$ where the constants A and B are chosen according to the size of the colon.
3. Finally a minimal path search is done using the centering potential P' . The final centered path is shown in Fig. 2-left in full line its virtual view is shown in Fig. 2-right.

3.4 End point detection

The wavefront expansion will stop when a user defined point has been reached. However, in practical applications it is better to automatically detect the end-point of the path for the following reasons :

- it reduces user interaction and therefore saves time and creates a user-independent result;
- a less experienced user may put the end point in the wrong location;
- the colon is often obstructed by remaining fluid, stool or colon contractions.

It is possible to detect when the propagating wavefront expands outside the colon lumen by simply checking the potential value of the voxel added to the

front. When this occurs, the entire colon lumen connected to the start point has been searched. The last voxel that was reached before expanding outside the colon lumen is the point that has the largest Euclidean distance (inside the colon) from the starting point, and is therefore chosen as automatic end point. The centering as explained in section 3.3 will be done between the starting point and the automatically detected end point.

3.5 Some results

Some results of the path tracker are shown in Fig. 3. We can clearly see the complex tortuous shape of the colon, and the quality of the resulting path.

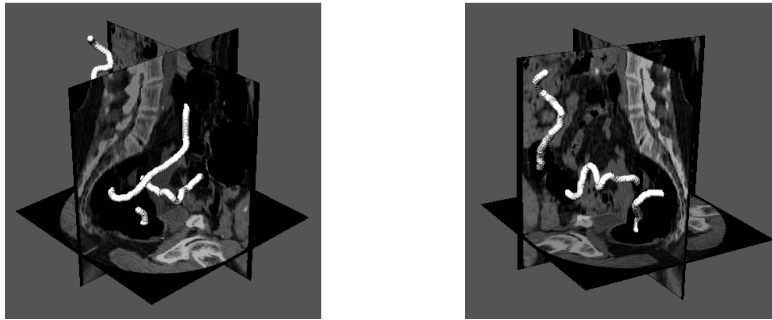


Fig. 3. Different views on a tracked path

4 Clinical study

4.1 Goal

A multi-user clinical study was performed using a prototype based on EasyVision (Philips Medical Systems). The purpose was to measure the speed and user-dependence of the automatic path tracker. To this aim, the path tracking tool has been evaluated by 5 different operators :

- two physicians with manual path tracking experience ($P1$, $P2$);
- two operators with abdominal anatomy knowledge but no virtual colonoscopy practice ($M1$, $M2$);
- one reference operator (R) familiar with the automatic path tracking tool.

As a comparison the user R also manually defined a path twice on the same dataset.

4.2 Data

Spiral CT data (5 mm slice thickness, 3 mm reconstruction interval) from 15 patients were used, corresponding to a total of 29 scans. During the patient preparation phase the colon was emptied as much as possible, and distended by inflating room air. In most cases, both prone and supine scanning was done.

4.3 Measurements

Time For each user, the wall clock time was measured using an automatic logging mechanism. In general, path construction consisted of following steps :

1. load and inspect data
2. place starting point in the cecum
3. track + center path
4. check result and modify/continue tracking if necessary

The time necessary to perform steps 3 and 4 were measured and both user interaction time and calculation time were taken into account.

User-dependence To measure the user-dependence of the path tracker, resulting paths P from different users were compared to the corresponding path R obtained by reference user in the following way :

1. **Warp path $P(i)$ and $R(j)$ to a common length parameter k**
We want to locally stretch and compress paths $P(i)$ and $R(j)$ using warping functions $w_P(k)$ and $w_R(k)$. The goal is to obtain the optimal warping satisfying

$$(w_P, w_R) = \underset{w_1, w_2}{\operatorname{argmin}} \left(\sum_k \|P(w_1(k)) - R(w_2(k))\| \right) \quad (5)$$

where all points of P and R are addressed by the warping. This mapping is calculated using the Dynamic Time Warp algorithm ([8]).

2. **Calculate Euclidean distances d between corresponding points**
After warping, the path distances d can be calculated as

$$d_{P,R}(k) = \|P(w_P(k)) - R(w_R(k))\| \quad (6)$$

and are shown in Fig.4 as a function of the common path length k .

3. **Extract the common part**

To exclude the effect of the exact start and end point, we only consider the common part of paths P and R , which is defined in Fig.4.

4. **Measure the similarity \mathcal{S} between the paths**

We propose to measure the similarity by using the percentage of the common path length k where the distance d is below a threshold t , written as $\mathcal{S}_{(P,R)}^t$.

The chosen measure for path similarity $\mathcal{S}_{(P,R)}^t$ is symmetrical, it is robust to a complete different choice in starting position, and it gives a more reliable result of similarity between paths than e.g. the maximum distance.

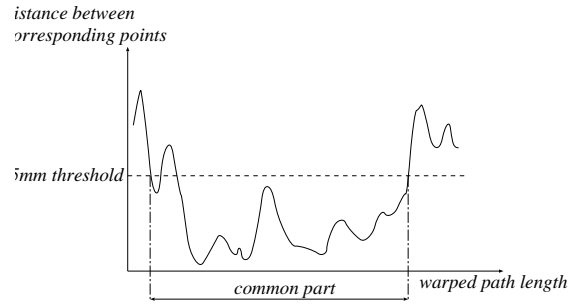


Fig. 4. The Euclidean difference between corresponding points plotted versus the warped path length k . The common part is defined as the longest possible stretch between two positions where the distance falls below the threshold of 5 mm.

Table 1. Timing results of the automatic path tracker.

time	P1	P2	M1	M2	R	average	manual
user time	2.8	4.6	5.3	4.5	2.0	3.6	30.0
calculation time	1.0	1.0	1.2	1.7	1.3	1.2	
total time	3.8	5.6	6.5	6.2	3.2	4.8	30.0

Time expressed in minutes per scan.

4.4 Results

Time The results of the time measurements are shown in Table 1. The average time needed for path tracking is 4.8 minutes per scan, measuring both user interaction time and calculation time.

No significant differences (95% confidence level, student t-test) were found between the experienced physicians ($P1$ and $P2$) and the other users ($M1$, $M2$), excluding the reference user R . These results are in agreement with a previously published study [4], where an average of 4.5 minutes was measured on 27 cases. The timing for the manual case has no statistical meaning, but is merely given for comparison.

User-dependence Table 2 summarizes the measurements of the path correspondence using the similarity measures \mathcal{S}^2 (2 mm threshold) and \mathcal{S}^5 (5 mm threshold) as defined in section 4.3.

On the average, an automatically defined path differs less than 5 mm from the reference over 94% of its length. Again, the results of the manual path tracking is given for comparison. Both manually tracked paths differed less than 5 mm over only 68% of their lengths.

Table 2. Correspondence with reference path.

correspondence	P1	P2	M1	M2	average	manual
\mathcal{S}^2	79%	89%	81%	92%	85%	20%
\mathcal{S}^5	89%	97%	93%	95%	94%	68%

Correspondence expressed in percentage of the path length where the reference path is closer than 2 mm (first row) or 5 mm (second row).

5 Conclusion

Automatic path tracking provides a fast and easy way to determine the colon centerline. The resulting path lies completely inside the colon, is as much centered as possible and is smooth.

The path tracker can be used by less experienced operators without significant differences in time or resulting centerline. This is an important result, since it allows to separate the path tracking task from the actual inspection task. The former task can be done as a preprocessing step by a different person since the results of the path tracker are largely operator-independent. Separating path tracking and inspection will increase the physician's efficacy, reduce the cost and allow a more widespread application of path-based navigation and visualization.

References

1. Cohen, L.D., Kimmel, R.: Global Minimum for Active Contour Models: A Minimal Path Approach. *International Journal of Computer Vision*. **24** (1997) 57–78
2. Deschamps, T., Cohen, L.D.: Automatic construction of minimal paths in 3D images for virtual endoscopy. *Computer Assisted Radiology and Surgery, CARS'99, Paris*. (1999)
3. Deschamps, T., Cohen, L.D.: Minimal Paths in 3D Images and Application to Virtual Endoscopy. *European Conference on Computer Vision, ECCV'00, Dublin*. (2000)
4. Rogalla P., Verdonck B., Truyen R., Hamm B.: Efficacy of automatic path tracking for virtual colonoscopy. *RSNA 1999*
5. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. *International Journal of Computer Vision*. **4** (1988) 321–331
6. Malladi, R., Sethian, J.A.: A Real-Time Algorithm for Medical Shape Recovery. *Proceedings of International Conference on Computer Vision*. (1998) 304–310
7. Sethian J.A.: *Level set methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision and Materials Sciences*. Cambridge University Press (1999)
8. Rabiner, L. R., Rosenberg, A. E., and Levinson, S. E.: Considerations In Dynamic Time Warping Algorithms For Discrete Word Recognition *IEEE Trans. On Acoustics, Speech and Signal Processing* , vol. ASSP-26, No. 6, pp. 575-582, December 1978.