# Unfolding of Virtual Endoscopy Using Ray-Template

Hye-Jin Lee, Sukhyun Lim, and Byeong-Seok Shin

Inha University, Dept. of Computer Science and Information Engineering
253 Yonghyun-dong, Nam-gu, Inchon, 402-751, Rep. of Korea
{jinofstar,slim}@inhaian.net, bsshin@inha.ac.kr

**Abstract.** Unfolding, one of virtual endoscopy techniques, gives us a flatten image of the inner surface of an organ. It is more suitable for a diagnosis and polyp detection. Most common unfolding methods use radial ray casting along with pre-computed central path. However, it may produce false images deformed and lost some information because adjacent ray planes cross when the organ's curvature is relatively high. To solve it, several methods have been presented. However, these have severe computational overhead. We propose an efficient crossing-free ray casting for unfolding. It computes ray-cones according to curvature of the path. Then in order to avoid intersection between ray-cones, it adjusts direction of ray-cones detected while testing intersection. Lastly, it determines direction of all rays fired from sample points between control points by simple linear interpolation. Experimental results show that it produces accurate images of a virtually dissected colon and takes not much time.

## 1 Introduction

Virtual endoscopy is a non-invasive inspection of inner structure of the human cavities using tomography images (e.g., CT and MRI). While an optical endoscopy is invasive and contains a certain degree of risk for patients, virtual endoscopy avoids the inconvenience of an optical endoscopy and improves the accuracy of diagnosis. However, virtual endoscopy cannot provide entire view of organ surface due to limited field-of-view. Also some polyps may be hidden from view because of extremely complex structure of organs and folds. So, we have to devise a variety of the visualization methods that do not have physical limitation in the virtual world.

Recently, new methods to visualize human cavities have been proposed [5],[9],[10]. These methods based on the efficient way to inspect the inner surface of organ would be to open and unfold it. The virtual dissection of organs is analogies to the real anatomical dissection. We can easily and intuitively recognize special features and pathologies of the organ. However, unfolding methods have a problem of missing or duplicating polyps and take a lot of time to solve the problem. Therefore, they are not adequate for real-time application.

In this paper, we propose an efficient unfolding method that solves the problem of missing or duplicating significant features and produces a flatten image in short time. It determines direction of rays using ray-templates, the set of pre-computed direction vectors according to curvature of the path. In order to avoid intersection between adjacent ray-cones, it adjusts the normal vector of crossing ray-plane. Then it determines direction of all rays fired from sample points between control points using simple linear interpolation. Experimental results show that it does not produce erroneous images and takes less time compared with the previous methods.

J.L. Oliveira et al. (Eds.): ISBMDA 2005, LNBI 3745, pp. 69–77, 2005.

In the next section, we review problems of conventional unfolding methods. Sect. 3 presents the main steps of the method in detail. Experimental results are presented in Sect. 4 and conclusions and future work are described in Sect. 5.

## 2 Problems of Conventional Unfolding Methods

In general, unfolding methods compute ray directions using radial ray casting along with a central path of an organ. To transform a curved path into a straight one, computed ray planes are arranged in a straight line. A flatten image is generated by searching intersection points of organ surface with rays, and computing color of the points.

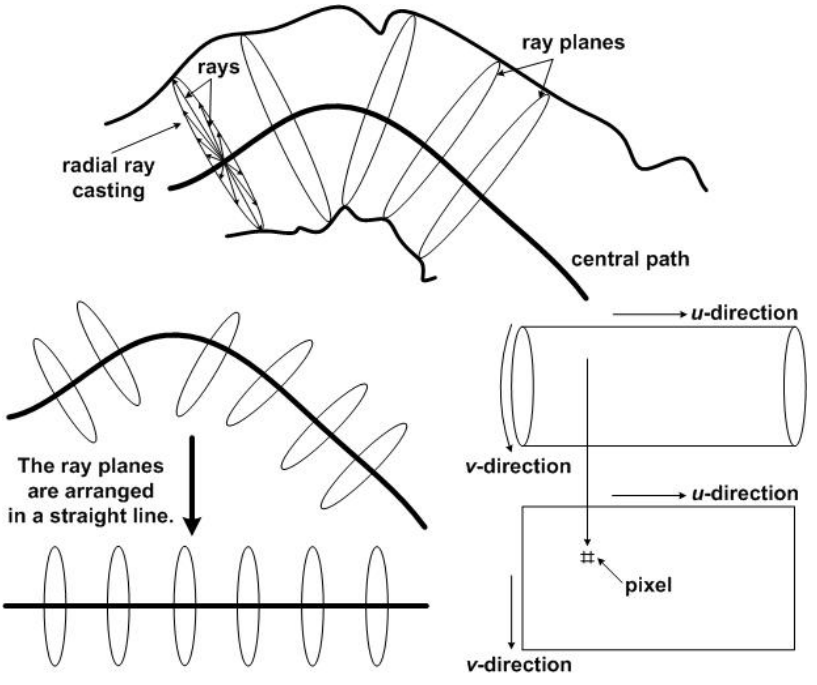


**Fig. 1.** Basic concept of unfold rendering. At first, it casts radial rays along with a central path and performs color composition along the ray directions. Then, it arranges ray planes on a straight line and applies mapping function to transform 2D image

These methods allow visualizing the complete surface of organs at once. Main problem of them is intersection of ray planes in high-curvature areas of the central path. In consequence, a polyp can appear twice in the flatten image or it can be missed completely (see Fig. 2).

To solve the intersection of ray plane, several methods have been presented. Wang et. al used electrical field lines generated by a locally charged path [5]. Electrical field lines that have the same polarity do not intersect each other. However, this method requires a lot of computations to calculate capacity for all electrical charges in the electrical field and to simulate the electrical force line.

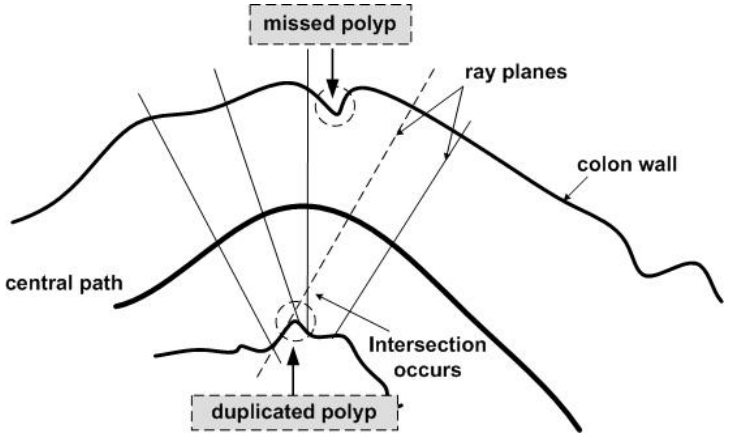


**Fig. 2.** Illustration of the missing and duplicating polyp in the conventional unfolding method due to intersections of the ray planes in high-curvature area. The ray plane represented as a dashed line produces a double polyp

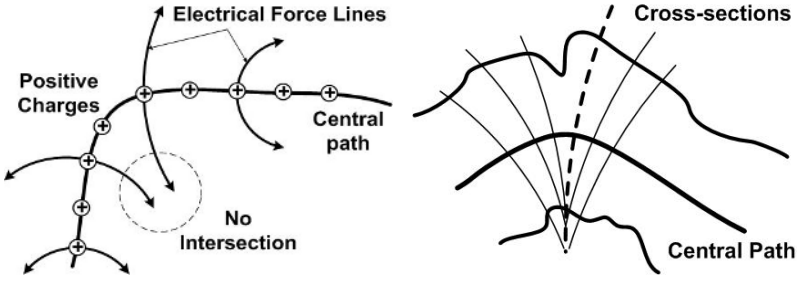


**Fig. 3.** Ray models proposed by Wang and Vilanova. Wang used electrical field lines generated by a locally charged path (left). Vilanova used a non-linear ray casting using the minimum distance of distance-map and vector field (right)

Vilanova et. al proposed a non-linear ray casting that prevents ray intersections by traversing non-linear rays using the minimum distance of distance-map and vector field[10]. It requires a lot of storages and long preprocessing time because it creates a distance-map and a vector field for traversing the minimum distance.

## 3 Unfold Rendering Using Ray-Template

We propose an efficient method to produce a flatten image without duplicating or missing features. In order to obtain the image of the virtually dissected organ, four steps have to be performed as shown in Fig. 4. We are given the central path of an organ cavity that represented as sample points. Control points are the part of sample points for taking shape of the path. Firstly, it classifies the path into two regions, curved regions and straight regions, according to the curvature of the central path. Then, it defines ray-cones only on control points. Secondly, it performs intersection test between two adjacent ray-cones and adjusts normal vector of a ray-cone when it recognizes the occurrence of intersection. Then, it computes direction of rays fired

from sample points using simply interpolation method. Lastly, it generates the entire unfold image by traversing rays with pre-calculated ray directions. It cannot offer an accurate image that identifies exact shapes of organ features. However, it can quickly generate images of excellent quality reflected important features.

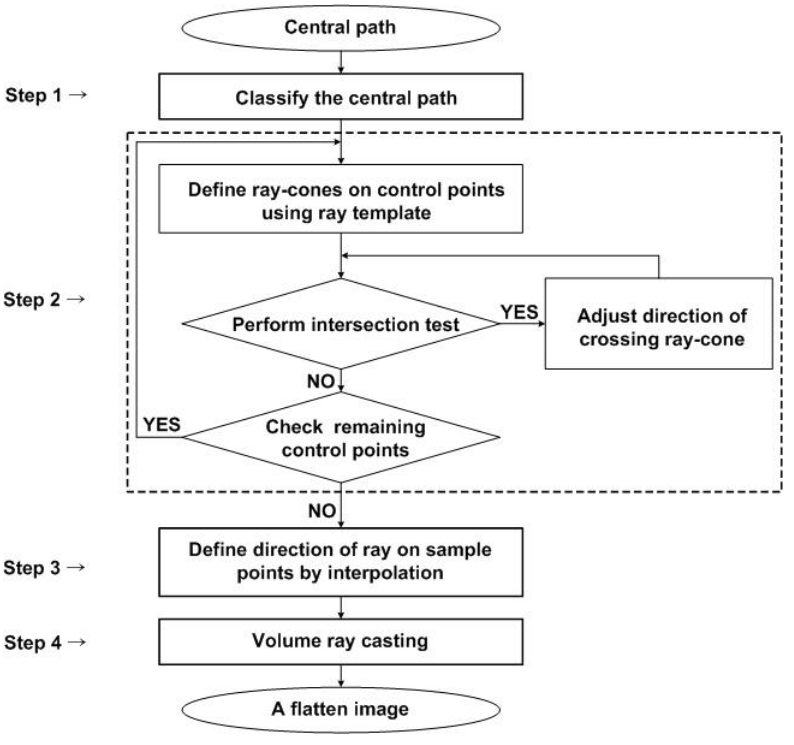


**Fig. 4.** A flow chart presents our procedure

### 3.1 Classification of the Central Path

A central path is composed of sampling points that correspond to $u$-coordinate. We classify it into curved regions and straight regions according to its curvature. After calculating dot product of a tangent vector $T$ of control point $P_i$ and that of adjacent control point $P_{i\text{-}1}$, it determines type of region. If the result of dot product is greater than pre-defined threshold($\varepsilon_1$), the type of region is regarded as straight. Otherwise, it is curved one.

*Ray template* (*RT*) is a set of ray vectors and its $k$-th ray is denoted as $RT(k)$. Radial and conic ray template mean that rays are fired in the shape of circle and cone form control points respectively. We apply a radial ray template to straight regions and a conic ray template to curved regions. Fig. 5 shows comparison of a radial template and a conic template. The ray plane given by a radial template **s** is crossing with adjacent ray. However, a conic template **t** can avoid intersection because its effect is similar to searching points that have minimum distance using distance-map. Fig. 6 shows a section of a ray-cone.

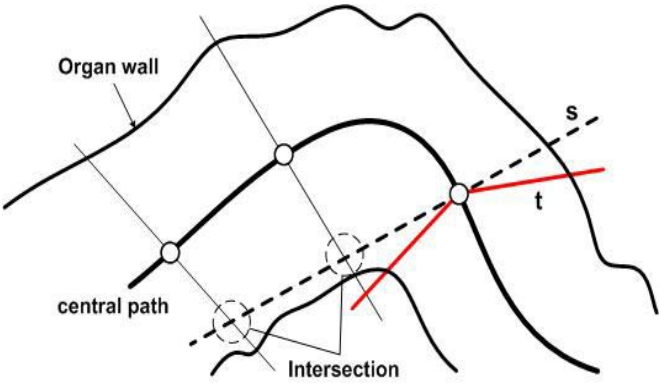


**Fig. 5.** Illustration of applying radial and conic templates on the path: **s** is a radial template, **t** is a conic template

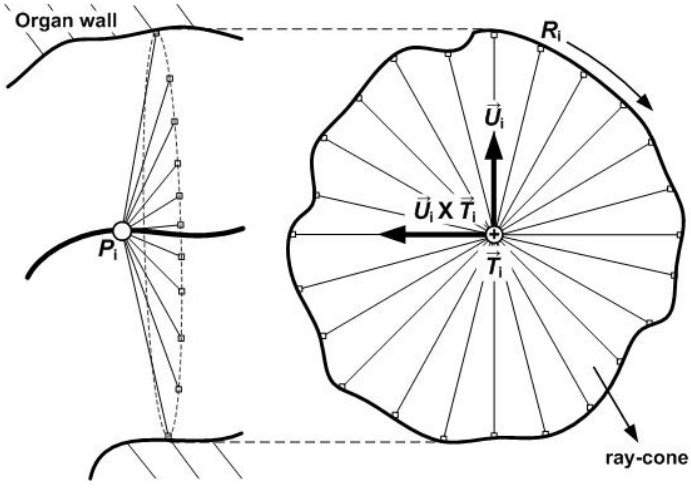


**Fig. 6.** Illustration of a section of a ray-cone $R_i$ on control point $P_i$

We assume that a central path $C(P_i)$ has control points $P_i$. It finds new basis that has a tangent vector of control point $P_i$ like as $z$-axis. Then it calculates direction of rays fired from control points by multiplying ray template $RT$ with transformation matrix. The advantage of basis transform is faster processing speed because it applies a transformation matrix to a lot of vectors in a lump. The point where a ray meets with the first non-transparent voxel is called *a boundary point*. A set of boundary points consisted of the outline of ray-cone. It derived from control point $P_i$ is $R_i$, the $k$-th point is $R_i(k)$.

### 3.2 Intersection Avoidance

Under occurrence of intersections, severe artifacts such as duplicating or eliminating features may come in a final image. So we have to check whether two consecutive ray-cones are intersected and adjust one of ray-cones when they are intersected.

We propose an intersection test using spatial coherence of shape of organ. At First, it computes vectors between boundary points that have identical index in adjacent ray-cones. It calculates dot product of the first vector $\vec{v}_1$ and the other vectors respectively. If results of dot product for all points are more than threshold ($\varepsilon_2$), the ray-cones do not intersect.

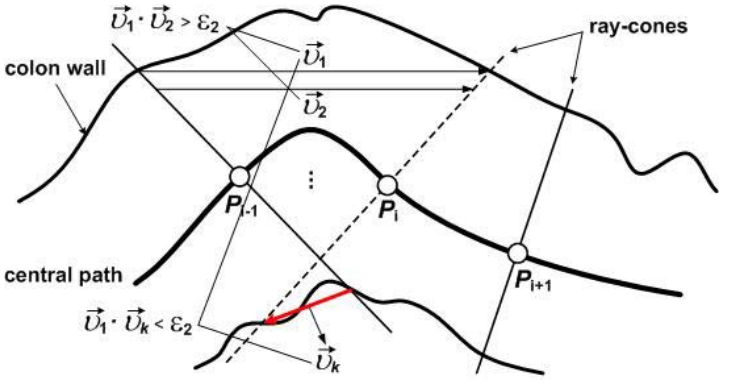


**Fig. 7.** Illustration of intersection test: Vector $\vec{v}$ is defined as direction vector that has identical index in two consecutive ray-cones

When intersection occurs, it adjusts the direction of current ray-cone $R_i$. The new normal vector of $R_i$ is determined by simply interpolating normal vectors of crossing ray-cones. It performs intersection test repeatedly until they do not intersect.

$$T'_i = \omega \times T_{i-1} + (1-\omega) \times T_i$$
$$\omega = \alpha \times k,\ 0 \le \omega < 1,\ k \text{ is the number of iteration.} \tag{1}$$

In equation 1, if $\omega$ is zero, the new normal vector is the same as the normal vector of neighboring ray-cone. Consequently, two consecutive ray-cones do not intersect because they are parallel to the each other.

### 3.3 Determination of Ray Direction Using Linear Interpolation

After determining ray-cones on control points, it computes direction of ray fired from sample points. It does not produce ray-cones for sample points instead of determining direction of rays using linear interpolation of boundary points.

As shown in Sect. 3.2, it computes vector $\vec{v}$ between boundary points that have the same index in adjacent ray-cones. Then, the direction of ray is computed using linear interpolation. We can compute the direction of each ray on sample point $S_n$ as following equation.

$$R_{S_n} = R_{i-1} + \vec{v} \times \frac{n}{m+1} - S_n \tag{2}$$

In above equation, $m$ is the total number of sample points between adjacent control points. The $n$ is current position of sample points between $P_i$ and $P_{i-1}$.

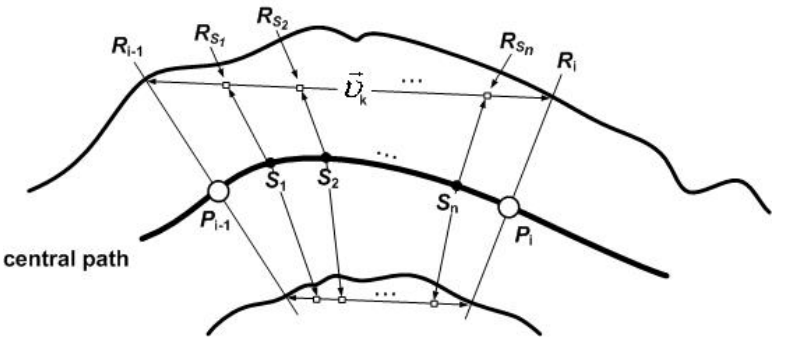


**Fig. 8.** Example of linear interpolation for direction of rays on sample points using ray-cones on control points. White circles are control points and black ones are sample points

### 3.4 Unfolding Image Generation

After determining ray directions, it performs color composition along the ray directions using ray casting.

As shown in Fig. 9, compositing color of each ray corresponds to each pixel on final image. The *u*-coordinate of an image is the index of sample points and the *v*-coordinate of an image is the index of rays.

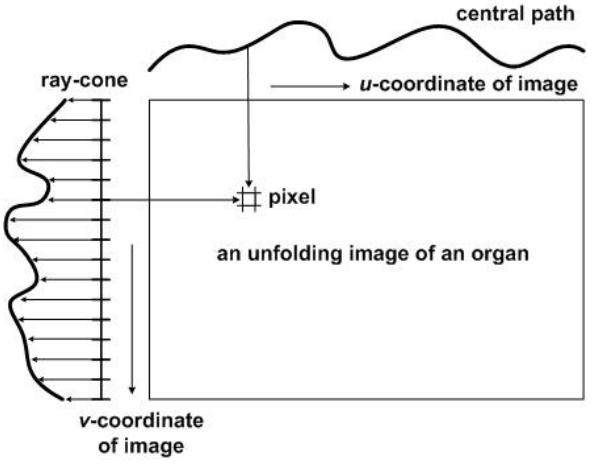


**Fig. 9.** Producing a flatten image by mapping ray-cones on 2D image

## 4 Experimental Results

In order to estimate the image quality and processing time of final image, we implemented basic radial ray casting and our method. These methods are implemented on a PC equipped with Pentium IV 3.0GHz CPU, 1GB main memory. The volume dataset is obtained by scanning a human abdomen with a multi-detector CT of which the resolution is 512×512×684. We measure the processing time in a human colon. It

uses ray-templates consist of 360 vectors at the interval of one degree and produces an image of N×360 resolution where N is the number of sample points.

**Table 1.** Comparison of the time of generating unfold images with basic radial ray casting and our method. To obtain unfold image, we use volume ray casting. Although our method takes some more time to avoid intersection, it takes time less than basic radial ray casting and image quality is much better

| # of sample points | Our method(sec) | Original method(sec) | Efficiency (%) |
|---|---|---|---|
| 1992 | 0.9484 | 4.3284 | 456 |
| 1972 | 0.9406 | 4.2222 | 449 |
| 1320 | 0.7359 | 2.9963 | 407 |
| 996 | 0.5861 | 2.2668 | 387 |
| 334 | 0.1828 | 0.7641 | 418 |

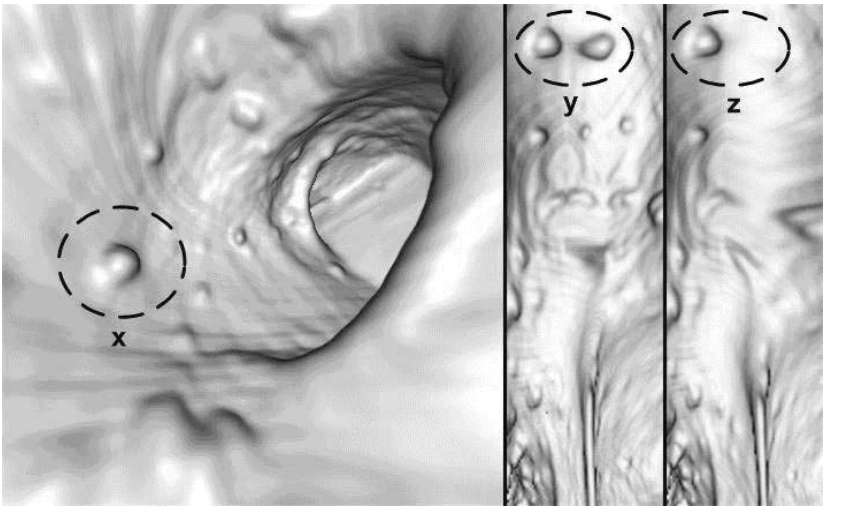


**Fig. 10.** Comparison of virtual endoscopy image (left) and unfold rendering images (middle, right) for the same region. We can find the fact that single feature is represented twice in the image produced by basic radial ray casting without intersection test (middle). However, a polyp represented in our unfold image is the same as that of the endoscopy image (right)

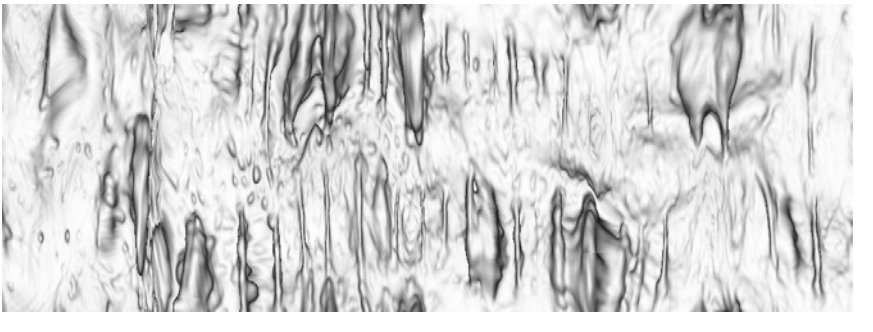

**Fig. 11.** Practical unfold image for human color in two different regions with 996×360 resolutions

## 5 Conclusions

We propose an efficient unfolding method to inspect the inner surface of the colon. It computes ray-cones using ray-templates according to curvature of the path, and adjusts ray-cones that do not meet with adjacent ray-cones using a simple intersection avoidance method. Lastly, it determines direction of all rays fired from sample points by simple linear interpolation. The presented approach solves the problem of missing or duplicating polyps and produces a flatten image in less time compared with the previous methods. Scaling method to make virtual dissection image that is analogies to the real anatomical dissection is the subject of future work. Acceleration of processing time is also the theme of future study.

## Acknowledgement

This work was supported by INHA UNIVERSITY Research Grant.

## References

1. Wang, G., Vannier, M.W.: GI tract unraveling by spiral CT. In Proceedings SPIE, Vol. 2434. (1995) 307-315
2. Grőller, E.: Nonlinear Ray Tracing: Visualizing strange worlds, The Visual Computer, Vol. 11. (1995) 263-274
3. McFarland, E.G., Brink, J.A., Balfe, D.M., Heiken, J.P., Vannier, M.W.: Central axis determination and digital unraveling of the colon for spiral CT colography. Academic Radiology, Vol. 4. (1997) 367-373
4. Wang, G., McFarland, E.G., Brown, B.P., Vannier, M.W.: GI tract unraveling with curved cross-sections. IEEE Transactions on Medical Imaging, Vol. 17, No. 2. (1998) 318-322
5. Wang, G., Dave, S.B., Brown, B.P., Zhang, Z., McFarland, E.G., Haller, J.W., Vannier, M.W.: Colon unraveling based on electrical field: Recent progress and further work. In proceedings SPIE, Vol. 3660. (1999) 125-132
6. Vilanova, A., Grőller, E., Kőnig, A.: Cylindrial approximation of tubular organs for virtual endoscopy. In Proceedings of Computer Graphics and Imaging (2000) 283-289
7. Haker, S., Angenent, S., Tannenbaum, A., and Kikinis, R.: Nondistorting Flattening Maps and the 3-D Visualization of Colon CT Images. IEEE Transactions on Bio-medical Engineering, Vol.19, No.7. (2000) 665-671
8. Haker, S., Angenent, S., Tannenbaum, A., and Kikinis, R.: Nondistorting Flattening for Virtual Colonoscopy. MICCAI (2000) 358-366
9. Vilanova, A., Wegenkittl, R., Kőnig, A., Grőller, E., Sorantin, E.: Virtual colon flattening. In VisSym '01 Joint Eurographics – IEEE TCVG Symposium on Visualization (2001)
10. Vilanova, A., Wegenkittl, R., Kőnig, A., Grőller, E.: Nonlinear Virtual Colon Unfolding. IEEE Visualization (VIS) (2001) 411-579
11. Zhang, Z.: Methods to Visualize Interiors of Human Colons in Volumetric Datasets. Medinfo (2004) 1931
12. Zhu, L., Haker, S., Tannenbaum, A,.: Flattening Maps for the Visualization of Multibranched Vessels. IEEE Transactions on Medical Imaging, Vol. 24, No. 2. (2005) 191-206