

# Virtual Fly-Over: A New Visualization Technique For Virtual Colonoscopy

M. Sabry Hassouna<sup>1</sup>, A. A. Farag<sup>1</sup> and Robert Falk<sup>2</sup>

<sup>1</sup> Computer Vision & Image Processing Laboratory (CVIP), University of Louisville,  
Louisville, Kentucky, 40292.

{msabry, farag}@cvip.uofl.edu

<sup>2</sup> Medical imaging, Jewish Hospital, Louisville, Kentucky.

robert.falk@jhhs.org \*

**Abstract.** In this paper, we propose a new visualization technique for virtual colonoscopy (VC). The proposed method is called *Virtual Fly-Over*, which splits the entire colon anatomy into exactly two halves. Then, it assigns a virtual camera to each half to perform fly-over navigation, which has several advantages over both traditional fly-through and related methods. First, by controlling the elevation of the camera, there is no restriction on its field of view (FOV) angle (e.g.,  $> 90^\circ$ ) to maximize visualized surface areas, and hence no perspective distortion. Second, the camera viewing volume is perpendicular to each colon half, so potential polyps that are hidden behind haustral folds are easily found. Finally, because the orientation of the splitting surface is controllable, the navigation can be repeated at a different split orientation to overcome the problem of having a polyp that is divided between the two halves of the colon. Quantitative experimental results on 15 clinical datasets have shown that the average surface visibility coverage is  $99.59 \pm 0.2\%$ .

## 1 Introduction

Colorectal colon cancer is the third most common form of cancer and the second leading cause of death among cancers in the western world [1]. Since colorectal cancer is largely preventable, the colonoscopy screening test is recommended for all people age 50 and over. Although colonoscopy detects more than 90% of colorectal cancers, it is invasive, uncomfortable, inconvenient, and sometimes can not reach the colon caecum, resulting in an incomplete exam [2]. On the contrary, virtual colonoscopy (VC) is a computer-based alternative to real colonoscopy. VC is not intended to replace real colonoscopy, but rather to complement it by providing additional supportive information such as visualizing in both directions, passing high grade stenoses, and planning for surgery. In addition, it is the only alternative offered to those patients that are not comfortable with real colonoscopy or are severely ill [3].

The common visualization technique that tries to simulate the real colonoscopy is the virtual fly-through navigation [4–6], where a virtual camera with a specific FOV

---

\* Acknowledgment: The colonoscopy dataset used in this paper were provided by 3DR Inc., Louisville, Kentucky.

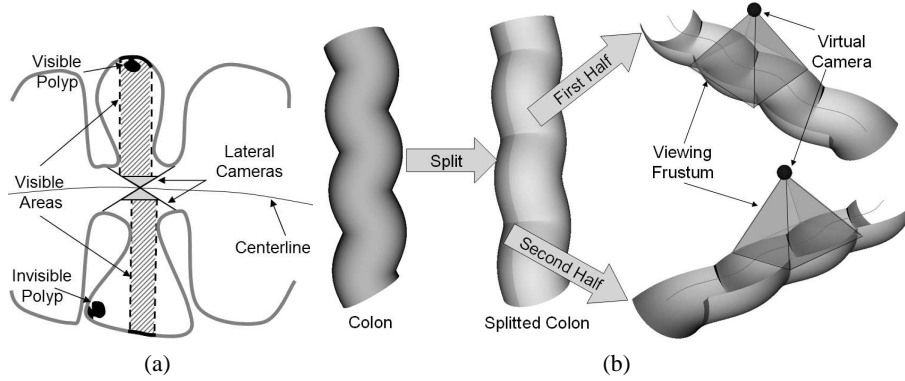
moves along a special planned path inside the colon to render its internal views. The direction of navigation either starts from the colon rectum side (antegrade), or from its caecum side (retrograde). In general, fly-through based-methods suffer from the following limitations: (1) the camera's FOV is limited, and hence results in a lower surface visibility coverage. (2) the navigation must be done in both antegrade and retrograde directions to maximize visualized surface areas, and hence it is very time consuming. Several visualization techniques have been proposed to overcome those shortcomings, which can be categorized as colon flattening and panoramic methods. The main idea behind colon flattening methods [7–9] is to initially transform the colon into a cylinder-like shape to reduce its overall tortuosity, and then to map the straightened colon onto a single image, which can be inspected from a single view point. Because flattening methods are based on geometric mapping, geometric distortion is inevitable. As a consequence, important diagnostic features can be altered or even eliminated.

Early work on panoramic methods was presented in [7]. The lateral side views of the colon have been captured by six views of  $60^\circ$  FOV each, which are then stitched together to form one view. The method has been extended in [10], where each centerline point is considered the center of a virtual cube. A virtual camera is located at that center to project the colon walls to each face of the cube using a  $90^\circ$  FOV. The cube is then unfolded into a single plane. One of the main drawbacks of this method is that a polyp may split among several faces of the cube. In addition, the layout of the unfolded cube is hard to follow, since a physician must observe six views simultaneously in two different directions (horizontal and vertical). Such drawbacks are partially corrected in [11]. The back face of the cube is removed, while the rest are mapped into a single plane. The front face is located at the middle of the plane, while its nearest neighbor square faces are mapped to trapezoidal windows. Due to this nonlinear mapping, the method introduces geometric distortion to 80% of the cube surface area. The common drawback of all panoramic methods is that polyps behind haustral folds can be easily overlooked under the following scenarios: (1) A polyp is located along the camera optical axis, and hence its depth perception is lost by a physician. (2) A polyp is completely blocked by a haustral fold as shown in Figure 1(a).

In this paper, we present the *virtual fly-over* as a new visualization technique for virtual colonoscopy, that overcomes the limitations of related methods.

## 2 Methods

The proposed method consists of two steps. In the first step, the colon is split along its centerline into exactly two halves, while in the second step, each half is assigned a virtual camera that performs virtual fly-over navigation as shown in Figure 1(b). In order to split the colon into exactly two halves, we have to use a polygonal clipping surface that passes through the colon's centerline. Clipping a 3D surface by a polygonal surface is a very tedious task. A better methodology is to approximate the polygonal surface by the union of finite size planes. Unfortunately, this simplification does not even work, because clipping must be accomplished by implicit surfaces such as a sphere, cube, cylinder, or an infinite plane. To overcome this problem, we divide the colon surface into consecutive strips of triangles (rings). Each ring is split into two halves by an



**Fig. 1.** (a) A schematic showing a cross section in a colon, where the lateral views of panoramic methods [10, 11] overlook a polyp that is completely blocked by a haustral fold. (b) Illustration of the proposed fly-over technique. Initially, the colon is split into two halves and then each half is assigned a virtual camera to perform fly-over navigation.

infinite plane that passes along its local centerline. Finally, each set of consecutive half rings are concatenated altogether to form one half of the colon. Initially the colon lumen is segmented using a seeded region growing algorithm. The centerline of the colon has been extracted using our recent work [4], which accurately and automatically computes the centerline of non-tubular structures. Finally, the colon's surface is reconstructed in 3D using the marching cubes algorithm [12].

### 1. Generation of Rings

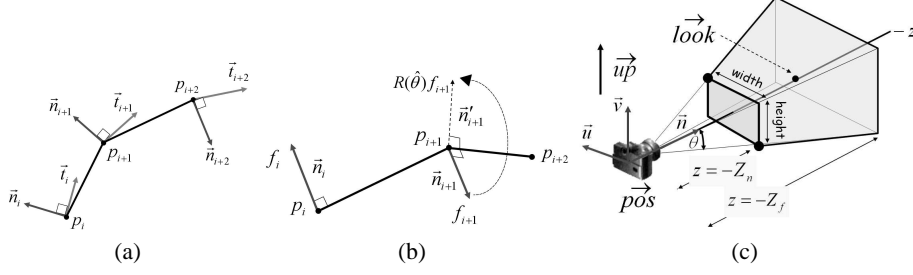
Let  $C$  be a smoothed version of the colon's centerline, which is sampled uniformly to  $N_p$  points as shown in Figure 2(a). Each point  $p_i$ , where  $1 \leq i \leq N_p$  is associated with a tangential vector  $\vec{t}_i$  and a normal vector  $\vec{n}_i$ . The centerline  $C$  is also divided into  $N_s$  segments  $l_j$ , where  $1 \leq j \leq N_s$ . Each segment  $l_j$  is  $1cm$  in length (i.e, 20 voxels for  $0.5mm$  voxel size). The points of each segment  $l_j$  are assigned a specific label based on the following *label* function.

$$\forall p_i \in l_j, \quad label(p_i) = j \quad (1)$$

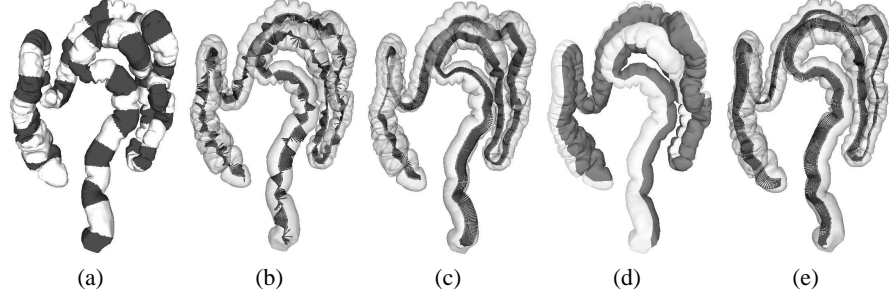
In order to divide the colon surface into consecutive rings, we use a surface skinning method, where each mesh triangle is assigned the label of the nearest centerline segment  $l_j$ . As a consequence, the set of surface triangles of the same label form a ring. Let  $c_k$  be the geometric center of each surface triangle  $r_k$ , where  $1 \leq k \leq N_r$  and  $N_r$  is the total number of surface triangles. Then, the skinning process is governed by the following equation.

$$\hat{p}_i = \arg \min_{p_i \in C} \|c_k - p_i\|^2, \quad label(r_k) = label(\hat{p}_i) \quad (2)$$

Surface skinning is a fast process of computational complexity  $O(N_r N_p)$ . In Figure 3(a), we show the generated rings of a clinical colon dataset.



**Fig. 2.** (a) The centerline  $C$  is uniformly sampled to  $N_p$  points. (b) Each point  $f_{i+1}$  is rotated around the line segment that is given by the points  $p_{i+1}$  and  $p_{i+2}$  until the rotation angle between the normal vectors  $\mathbf{n}_{i+1}$  and  $\mathbf{n}_i$  is minimized. (c) The pinhole virtual camera model.



**Fig. 3.** (a) Colon surface is divided into consecutive rings ( $l_j > 1cm$ ). (b) The computed normals along the centerline may change their directions abruptly. (c) Aligned normals provide a good approximation of the clipping surface. (d) Splitting colon surface into two halves. (e) Changing the direction of the first normal vector results in a new split orientation surface.

## 2. Splitting Rings

Each ring  $R_j$  spans multiple centerline points, whose starting and ending points are  $q_j$  and  $q_{j+1}$ , respectively. In order to split  $R_j$  into two halves, we clip it using an infinite plane  $\pi_j$ , whose center is  $q_j$  and whose normal vector is given by

$$\vec{u}_j = (q_{j+1} - q_j) \times \vec{n}_j \quad (3)$$

To generate the entire clipping planes along all rings, we have to first compute the normal vectors along the centerline, which can be generated using a common method in computer graphics [13]. The normal vector  $\vec{n}_i$  at each centerline point  $p_i$  is given by,

$$\vec{w}_i = \vec{up} \times \vec{t}_i, \quad \vec{n}_i = \vec{t}_i \times \vec{w}_i \quad (4)$$

where,  $\vec{up}$  is an arbitrary vector, which is adaptively chosen at each point to be any of the Cartesian basis vectors as long as it is not parallel to  $\vec{t}_i$ . Since we approximate the polygonal splitting surface by the union of finite size planes, then

each two consecutive planes must share an edge, which implies that each two consecutive normal vectors  $\vec{n}_i$  and  $\vec{n}_{i+1}$  are in the same plane. Unfortunately, due to the high tortuosity nature of the colon and hence its centerline, the generated normal vectors along the centerline using Eq. (4) may change their directions abruptly several times as shown in Figure 3(b). To overcome this problem, we propose the following method. Let  $\vec{n}_i$  be described by the end points  $p_i$  and  $f_i$ , while  $\vec{n}_{i+1}$  by  $p_{i+1}$  and  $f_{i+1}$  as shown in Figure 2(b). Each point  $f_{i+1}$  is rotated around the line segment that is given by the points  $p_{i+1}$  and  $p_{i+2}$  until the rotation angle between  $\vec{n}_{i+1}$  and  $\vec{n}_i$  is minimized, which can be formulated mathematically by the following optimization problem,

$$\hat{\theta} = \arg \min_{\theta} \|f_i - R(\theta)f_{i+1}\|^2 \quad (5)$$

where  $R(\theta)$  is the rotation matrix. In Figure 3(c), we show the aligned normal vectors of the same colon dataset of Figure 3(b), while in Figure 3(d), we show the results of splitting the same dataset into exactly two halves.

### 3. Controlling Split Orientation

Since each normal vector  $\vec{n}_{i+1}$  is aligned with respect to the previous vector  $\vec{n}_i$ , then by changing the direction of the first normal vector  $\vec{n}_1$ , the rest of the normal vectors will change their orientations as well. Therefore,  $\vec{n}_1$  controls the orientation of the split surface. The first normal vector  $\vec{n}_1$  of Figure 3(c) has been rotated by  $90^\circ$  to yield a new split orientation surface as shown in Figure 3(e).

### 4. Virtual Fly-Over

Each colon half is assigned a perspective virtual camera, whose model is shown in Figure 2(c). The part of the pyramid that is enclosed by the clipping planes  $Z_n$  and  $Z_f$  is called the viewing volume or frustum. The orientation of each camera is controlled by four parameters that are computed automatically during fly-over navigation. The parameters are the camera position vector  $\vec{pos}$ , the look-at vector  $\vec{look}$ , which describes the direction of projection, the view-up vector  $\vec{up}$ , and the field of view angle  $\theta$ . The look-at vectors of the first half ( $l = 1$ ) and second half ( $l = 2$ ) are given by,

$$\vec{look}_{i,l=1} = \vec{t}_i \times \vec{n}_i, \quad \vec{look}_{i,l=2} = -\vec{look}_{i,l=1} \quad (6)$$

The camera position and view-up vectors of each half are given as follows,

$$\vec{pos}_{i,l} = p_i + h_i \frac{\vec{look}_{i,l}}{\|\vec{look}_{i,l}\|}, \quad \vec{up}_{i,l} = \vec{t}_i \quad (7)$$

where  $h_i$  is the camera elevation from each centerline point  $p_i$ . According to Figure 2(c), the camera's FOV is given by,

$$\tan\left(\frac{\theta}{2}\right) = \frac{height}{2Z_n} \quad (8)$$

During centerline extraction, each point  $p_i$  encodes its distance from the colon's boundary  $D(p_i)$ . By making  $p_i$  the focal point of the camera, where the image

plane is located, then the elevation of the camera  $h_i$  at each point  $p_i$  that maximizes surface visibility coverage, while maintaining the same angle  $\theta$  is given by,

$$h_i = \frac{D(p_i)}{\tan(\theta/2)} \quad (9)$$

According to Eq. (9), there is no limitation on the FOV angle that maximizes surface visibility coverage. For example, for  $\theta = 90^\circ$ ,  $h_i = D(p_i)$ , while for  $\theta = 60^\circ$ ,  $h_i = 1.73D(p_i)$ .

### 3 Quantitative Validation

To quantitatively validate the proposed fly-over navigation method in terms of effectiveness (surface visibility coverage), clinical datasets of 15 patients were acquired using Siemens Sensation CT scanner. The dataset volume is  $512 \times 512 \times 580$  with voxel size  $0.74 \times 0.74 \times 0.75 \text{mm}^3$ .

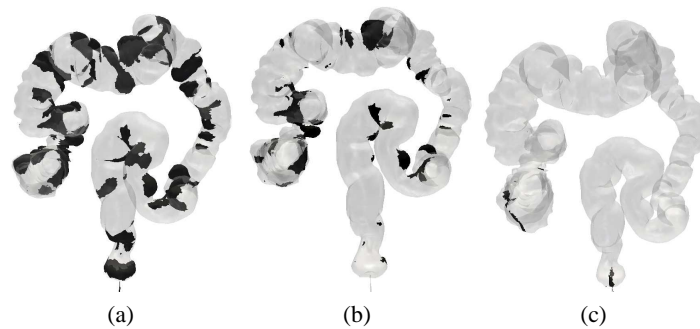
In Table 1, we compare the surface visibility coverage of the proposed method with the traditional fly-through method for three clinical datasets, while navigating in the antegrade, retrograde, and both directions at  $90^\circ$  FOV.

**Table 1.** Comparison of surface visibility by the fly-over and fly-through methods.

Navigation Direction	Navigation Technique	Surface Visibility (%)		
		Dataset A	Dataset B	Dataset C
Antegrade	Fly-through	79.29	78.95	79.38
	<i>Fly-over</i>	<b>99.35</b>	<b>99.68</b>	<b>99.33</b>
Retrograde	Fly-through	77.89	75.74	79.00
	<i>Fly-over</i>	<b>99.6</b>	<b>99.91</b>	<b>99.68</b>
Antegrade + Retrograde	Fly-through	93.64	93.12	93.38
	<i>Fly-over</i>	<b>99.7</b>	<b>99.95</b>	<b>99.73</b>

### 4 Discussion and Conclusion

The validation experiment 15 clinical datasets have shown that the surface visibility coverage by the proposed method irrespective of the navigation direction is on average  $99.59\% \pm 0.2$ . To ensure 100% surface coverage, invisible areas are clustered using a connected component algorithm and then properly visualized. Although panoramic methods have reported nearly the same surface visibility coverage [10, 11], they still suffer from other limitations as mentioned earlier. The increase in surface visibility coverage by the proposed fly-over method, while navigating in both directions is insignificant, which was expected because the camera viewing volume is perpendicular to each colon half. Therefore, fly-over navigation in either direction (antegrade or retrograde) is sufficient for nearly complete surface coverage. In Figure 4, we superimpose



**Fig. 4.** Comparison of surface visibility coverage by (a) Fly-through in antegrade direction, (b) Fly-through in both directions, (c) Fly-over in antegrade direction.

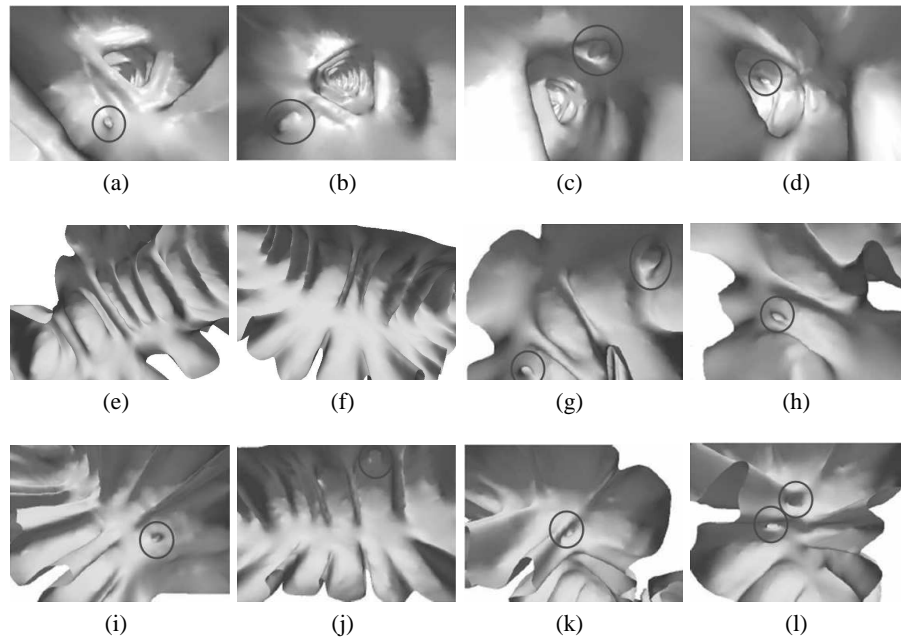
on one clinical dataset undetected surface patches (dark color) by both the fly-through and the proposed method. Notice that fly-over method greatly reduces blind areas.

In the first row of Figure 5, we show the rendered views during traditional fly-through navigation, while in the middle and last rows, we show the rendered views during fly-over navigation of the first and second colon halves, respectively. While some rendered views of both fly-through and fly-over are showing the same number of polyps as shown in the first two columns of Figure 5, some rendered views by the proposed fly-over method are showing more polyps than the corresponding fly-through views as shown in the last two columns of the same figure. The evaluation time was on average 26 minutes by fly-through method in both directions and 13 minutes (including 3 minutes for surface reconstruction, skinning, splitting, and rendering) by fly-over method on a 3Ghz Pentium computer with 2 GB memory and nVidia Quadro FX 3000 graphics card.

To conclude, in this paper we have presented a new visualization technique for virtual colonoscopy, whose advantages can be enumerated as follows. Due to the nature of the fly-over navigation, the viewing volume is perpendicular to each colon half, which results in higher surface visibility coverage and higher sensitivity. By controlling the elevation of the camera, there is no restriction on its FOV angle to maximize visualized surface area, and hence the proposed method does not suffer from perspective distortion. Because the split orientation is controllable, the navigation can be repeated at a different split orientation if a polyp is divided between the halves of a colon. Follow-up research by our group includes large scale patient screening study to assess the proposed approach versus related methods, and to design a CAD system for polyp detection.

## References

1. Abbruzzese, J., Pollock, R.: *Gastrointestinal Cancer*. 1 edn. Springer (2004)
2. Macari, M., Berman, P., Dicker, M., Milano, A., Megibow, A.: Usefulness of ct colonography in patients with incomplete colonoscopy. *J. Roentgenol* **173** (1999) 561–564
3. Baert, A.L., Sartor, K.: *Virtual Endoscopy and Related 3D Techniques*. Springer (2001)



**Fig. 5.** Colon rendered views. (a-d) Fly-through navigation. (e-h) First colon half using fly-over. (i-l) Second colon half using fly-over.

4. Hassouna, M.S., Farag, A., Falk, R.: Differential fly-throughs (dft): A general framework for computing flight paths. In: MICCAI, Palm Springs, California, October 26-29 (2005)
5. Bouix, S., Siddiqi, K., Tannenbaum, A.: Flux driven fly throughs. In: Computer Vision and Pattern Recognition. (2003) 449–454
6. Kang, D.G., Ra, J.B.: A new path planning algorithm for maximizing visibility in computed tomography colonography. *24*(8) (2005) 957 – 968
7. Paik, D., Beaulieu, C., et al.: Visualization modes for ct colonography using cylindrical and planar map projections. *Journal of Computer Tomography* *24*(2) (2000) 179–188
8. Haker, S., Angenent, S., Tannenbaum, A., Kikinis, R.: Nondistorting flattening maps and the 3d visualization of colon ct images. *19*(7) (2000) 665–671
9. Oda, M., Kitasaka, T., Hayashi, Y., Mori, K., Suenaga, Y., ichiro Toriwaki, J.: Development of a navigation-based cad system for colon. In: MICCAI. (2005) 696–703
10. Vos, F.M., van, R.E., Serlie, I.W.O., et al.: Three-dimensional display modes for ct colonography: conventional 3d virtual colonoscopy versus unfolded cube projection. *Radiology* *228* (2003) 878–885
11. Geiger, B., Chafd'hotel, C., Sudarsky, S.: Panoramic views for virtual endoscopy. In: MICCAI. (2005) 662–669
12. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: In Proc. of SIGGRAPH. (1987) 163–169
13. Hill, F.J.: Computer Graphics Using OpenGL. Prentice Hall PTR (2000)