An Interactive Virtual Endoscopy Tool

Delphine Nain [1] Steven Haker [2] Ron Kikinis [2] W. Eric L. Grimson [1]
[1] Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge MA, USA.
delfin@ai.mit.edu, http://www.ai.mit.edu/people/delfin
[2] Brigham and Women's Hospital, Harvard Medical School, Boston MA, USA.

1 Introduction

In this paper, we present the design and implementation of a 3D Virtual Endoscopy system for facilitating diagnostic and surgical planning phases of endoscopic procedures. Our system allows the user to interactively explore the internal surface of a 3D patient-specific anatomical model and to create and update a fly-through trajectory through the model to simulate endoscopy. We also present an automatic path planning algorithm that has been used to produce a colon fly-through with our system. We integrated our program in the 3D Slicer, a medical visualization program created at the MIT AI Lab in collaboration with the Surgical Planning Laboratory at the Brigham and Women's Hospital [1]. It provides the surgeons with a single environment to generate 3D models, use quantitative analysis tools and conduct a virtual endoscopy. The Virtual Endoscopy program has been used for virtual colonoscopy and cardiovascular projects.

In the next section, we present the motivation for interactive virtual endoscopy and the related work in the field. In Section 3, we describe our Virtual Endoscopy system. In Section 4, we present an automatic path planning algorithm that produces center points used by our system to fly-through a colon dataset. We conclude the paper with a discussion and evaluation of results.

2 Motivation for Interactive Virtual Endoscopy

2.1 Background

In this section, we present existing alternatives for examining the internal surface of a patient's anatomy. One approach is to introduce an endoscope, a device consisting of a tube and an optical system, through natural orifices or through incision in a hollow organ. The operator can view the inner surface of the organ using video-assisted technology, but does not have any information about the extent of the detected lesions or the anatomy beyond the walls of the organ.

Alternatively, Computed Tomography (CT) scans and Magnetic Resonance Imaging (MRI) provide cross sectional images of the interior of the body, that contain information not available to the endoscope, such as information on tissue shape through and beyond the walls of the organ. One drawback of this method is that the surgeon has to mentally align and combine the contiguous slices in order to perform a diagnosis and the resolution of the scans is much lower than the video image seen during a real endoscopy. A third approach is to use 3D medical visualization computer programs that create 3D models from 2D CT or MRI data [1, 2].

A virtual endoscopy tool can combine the strengths of the previous alternatives described above. It provides a technique for interactive exploration of the inner surface of the 3D model while being able to track the position of the virtual endoscope relative to the 3D model. Furthermore, the surgeon can record a trajectory through the organ for later sessions and save a movie to show other experts. If virtual endoscopy is combined with other tools that color-code the 3D models to highlight suspicious areas [3, 4, 5], the surgeon can have access to additional information during the diagnostic phase.

Virtual endoscopy also has some limitations. The cost to acquire CT and MRI data is higher than that of conventional endoscopy and the low resolution of images limits the resolution of the 3D model. The creation of a 3D model can be time consuming if the segmentation of anatomical structures is done manually. Finally, the use of 3D software can be non-intuitive and frustrating for novice users. Despite those limitations, clinical studies have shown that virtual endoscopy is useful for surgical planning by generating views that are not observable in actual endoscopic examination and can therefore be used as a complementary screening procedure and as a control examination in the aftercare of patients [6, 7, 8, 9, 10].

2.2 Related Work

The Virtual Endoscopy Center at Wake Forest University School of Medicine has created a Virtual Endoscopy System called FreeFlight [11] that explores 3D surface models. The surface models are created by automated image segmentation specific to each organ based on thresholding and region growing. In FreeFlight, the user navigates inside the surface model by choosing the motion of the camera with a mouse click (forward, no movement, backward) and rotates the camera up and down or left and right with the mouse motion. We have found that a user rarely moves the mouse in a perfect vertical or horizontal motion, so it is hard to only rotate up or rotate only to the right with such commands. Our 3D Gyro described in Section 3 allows the user to navigate with a single degree of freedom at a time for a more precise motion. In FreeFlight the user moves the camera in reference to what she sees on the endoscopic view. We have found that in order to navigate in a full 3D scene, it is also useful to have an actor representing the endoscopic camera in the scene with the anatomy and to have an intuitive way to move this actor with six degrees of freedom along its own coordinates or along the absolute coordinates of the 3D scene.

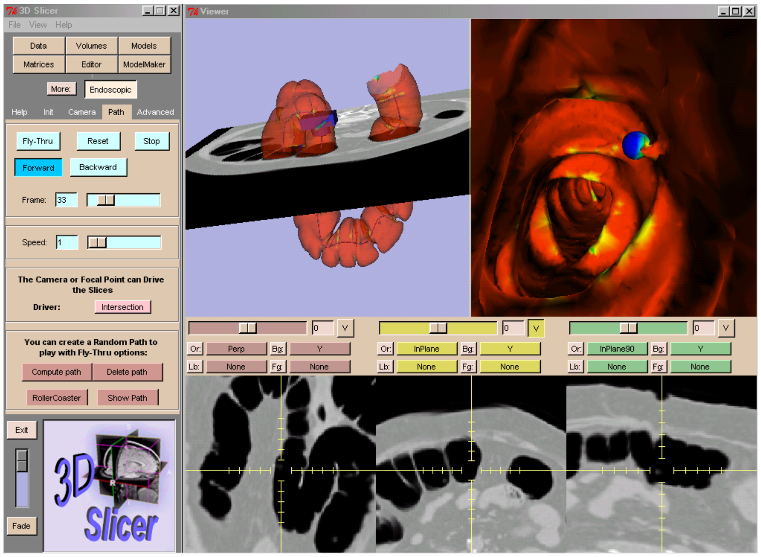
The Multidimensional Image Processing Lab at Penn State University [12] and The Visualization Laboratory at the State University of New York at Stony Brook [13] have developed virtual endoscopy programs (Quicksee) in volumetric environments with fast rendering rates. These programs have both a manual mode where the user can explore the scene interactively and an automatic mode where the path is pre-defined. In interactive mode, the user interface is similar to the one of the FreeFlight program and therefore has the same limitations. Another important feature not present in FreeFlight or Quicksee is the ability to interactively update the path once it has been created. As fast volume rendering becomes a feature of the 3D Slicer, our virtual navigation program will also be able to explore volumetric scenes. For now the 3D Slicer uses mainly surface rendering in order to run on multiple platforms on computers with non-specialized hardware.

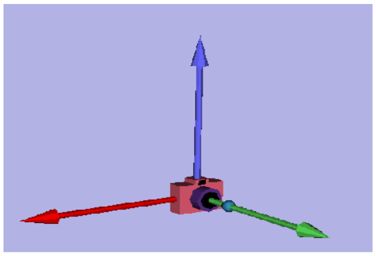
The Virtual Endoscopy Software Application (VESA) developed at the GE Corporate Research and Development Center is a prototype that has been developed in collaboration with the Surgical Planning Laboratory and used for many clinical studies [14, 15, 16, 4]. Many of its concepts have been used in the development of our system.

3 Virtual Endoscopy System

3.1 System Overview

Our goal in this work was to combine the strengths of non-invasive 2D imaging techniques with intuitive real-time 3D visualization. Our program simulates the surgical environment and provides the user with many navigational and path creation options as well as cross-references between windows.





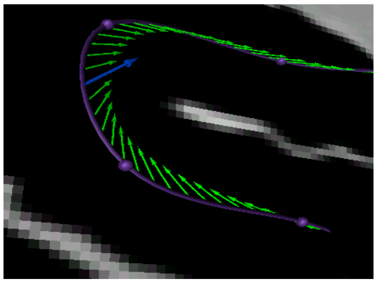


Figure 1. (left) The five cross reference screens during a Virtual Colonoscopy.
Figure 2. (top right) The user can move the endoscope intuitively with the 3D Gyro.
Figure 3. (bottom right) Path Creation: the path (purple tube) fits the user-defined *Landmarks* (purple spheres) on a grayscale. The green vectors defined at interval points of the path indicate the view direction of the endoscope. Users can change the orientation of any vector after selecting it (a selected vector is shown in blue).

3.2 System Environment

The Endoscopic module is part of the 3D Slicer that utilizes the Visualization Toolkit (VTK 3.2) for processing and the Tcl/Tk scripting language for the user interface. A 3D surface model can be generated from a set of grayscale volumetric images obtained by medical imaging scanners (i.e. CT scan or MR imaging) [17]. The grayscale images are displayed as 2D slices and can be segmented manually, semi-automatically or automatically with the 3D Slicer's suite of editing tools.

3.3 Display

The display of our Virtual Endoscopy system is based on five cross-referenced screens (Figure 1). In this Section we present those five screens as well as the different options to control the movement of the endoscope.

Screens The *World View* displays the main 3D scene that contains the anatomical models and the endoscope (top left screen of Figure 1). The endoscope is represented graphically by a camera. The user can navigate around the World View with the mouse.

The *Endoscopic View* is the screen that displays what the virtual endoscope is filming in real time (top right screen of Figure 1). The user can also navigate around this view with the mouse. This automatically updates the position and orientation of the endoscope in the World View.

In addition, three 2D *Slice Windows* show a cross section of the original grayscale data, called a *slice* (bottom three screens of Figure 1). If the user wants to track the position of the endoscope on the original grayscale data, the 2D windows are updated in real time and the position of the endoscope is shown with a cross-hair. The 2D slices can also be displayed in the 3D scene along with the models and the endoscope (Figure 1). Those cross reference features give the surgeon a global context to track the endoscope's position and interpret the grayscale data.

Navigation Moving and orienting the endoscope should be intuitive so that the user does not spend too much time trying to navigate through the model. We have investigated many ways to navigate the endoscope interactively.

One way to move the endoscopic camera is for the user to click and move the mouse over the endoscopic screen to rotate and move the virtual camera. The position and orientation of the endoscope actor is updated simultaneously in the World View. We have found that the motion of the mouse gives too many degrees of freedom for a precise translation or rotation of the endoscope. For example, to rotate the endoscope to the right, the user needs to move the mouse in a perfect horizontal line, otherwise the endoscope will also rotate up or down.

To choose exclusively which axes to use for translation or rotation, we have created six sliders that each control a degree of freedom (left/right, up/down, forward/backward for translation and rotation). Next to the sliders, the user can also read the world coordinates and angles of the endoscope and change them by hand. However, the use of sliders can be frustrating since the user has to shift her attention between the screen and the control panel to change the direction of motion.

We have found that one of the most intuitive navigation tool is a *3D Gyro* (Figure 2). The gyro is represented in the World View as three orthogonal axes of different colors that are 'hooked' to the endoscope. The user selects one of the axes by pointing at it with the mouse, pressing a key on the keyboard to choose between rotation or translation and then moving or rotating the endoscope along the active highlighted axis by moving the mouse. The 3D Gyro provides six degrees of freedom, but it enables the user to control each degree of freedom separately for a more precise control over the translational or rotational movements of the endoscope.

There are 2 modes of translation and rotation for the endoscope. In *absolute mode*, the endoscope is translated and rotated with respect to the world's axes. The axes of the 3D Gyro makes that notion intuitive to the user by staying aligned with the world axes. In *relative mode*, the translation and rotation are about the endoscope's own axes. In this mode, the 3D Gyro rotates along with the endoscope. As an additional feature, the user can pick a point on a model or on a cross-sectional slice (in the World View or in a 2D window) and choose to position the endoscope at that world coordinate. This enables the user to pick suspicious points on a grayscale slice, for example, and to move the camera to that point to explore the 3D model at that position.

Reformatted Slices By default, the three slices are oriented orthogonally (Axial, Sagittal and Coronal orientation) but the user has the option to reformat the slices in real-time. Each time the endoscope is moved or rotated, the orientation of the 3 slices can correspond to the current orientation of the endoscope's axes. Furthermore, the user has three options to determine the point where the 3 slices intersect, the *center* of the slices: the position of the endoscope, the position of the focal point, or a point picked on the surface of a model (in Figure 1, the slice displayed in the World View is centered at the polyp shown in blue in the Endoscopic View and oriented parallel to the lens of the endoscope).

Additional Features To simulate the surgical environment, we added an option to detect collision between the endoscope and the surface of the models. We check to see if the ray perpendicular to the view plane of the endoscope intersects the surface of a model and detect collision when the distance is under a threshold set by the user. To simulate a wide-lens endoscope, the user has the option to change the zoom factor as well as the view angle of the virtual endoscopic camera. In addition, the user is able to change the opacity of the model in the World View without affecting the opacity of the model in the Endoscopic View. This allows the user, for example, to track the position of the endoscope in the World View when it is inside a model, by making the model semi-transparent, while keeping the model at its full opacity in the Endoscopic View in order to see its internal surface.

3.4 Virtual Fly-Through

Path Generation To define a fly-through trajectory, the user has the option to specify both the position and the orientation of the endoscope at selected key points, called *Landmarks*. The first way to create a landmark is to record the current endoscope orientation and position. The second way is to pick a pixel on a grayscale (in the 2D or 3D View) which will add a landmark at the corresponding world position with a default orientation tangent to the path (looking "straight" ahead in the direction of movement).

As the user creates landmarks, the program interpolates between them using a cardinal cubic spline curve. We chose this interpolation function because it has second-order continuity, i.e the rate of change of the tangent vectors is matched at the landmarks. This ensures smooth transitions in the motion of the endoscope [18]. The path is represented as a tube with 3D vectors at interval points (Figure 3). Each vector shows how the endoscope will be oriented at that location of the fly-through. The user can select any vector and re-orient it to look in a different direction. The user can also select, move or delete any landmark. The path is updated and re-rendered in real time after any landmark or vector operation for the user to visualize the result.

Once the user is satisfied with the trajectory, she can start a fly-through of the model. A fly-through consists of a series of consecutive frames where the endoscope moves through each consecutive point on the spline. The user can play all the frames at a chosen speed, step through each frame manually, or jump to any frame number.

Saving Data At any time, the user can save a data file with the positions of the existing landmarks as well as the model properties. The format used by the slicer is the MRML (Medical Reality Modeling Language) format [1]. When a user-created data file is loaded, our system automatically creates a path from any landmarks defined in the file. The user can also save all frames during a fly-through to make a movie.

4 Calculating the Flight Path

Our system also provides for the automatic calculation of the flight path, which the user can then modify if desired. Here, we describe the method by which center points of a surface model are obtained. The center points are then used as landmarks to produce the flight path. Our starting point is a triangulated surface model of the colon, of the kind obtained through the use of the Marching Cubes algorithm [19], or other similar isosurface extraction algorithm. Here, it is assumed that the surface has a tubular topology, such as that of the colon. For branching surfaces such as vessels, similar approaches may be applied [20].

Our method is based on the following physical model. Let Σ denote the tubular colon surface with open ends described by two closed space curves σ_0 and σ_1. We suppose that these boundary curves are held at a constant temperature of 0 and 1 degrees respectively, and seek the steady-state distribution of temperature u across the surface. The standard theory of partial differential equations [21] tells us that this temperature distribution will smoothly vary between 0 and 1 degrees from end to end, and will be free of local maxima and minima away from the boundary curves. In fact, the function u will be harmonic, i.e. will satisfy Laplace's equation $\Delta u = 0$, and each level set $u^{-1}(t), t \in [0,1]$ will consist of a loop around the colon surface. Our flight path is then formed by the centers of mass of these loops.

The numerical method used to find the temperature distribution function is based on finite element techniques [22]. In [3] and [23], there are closely related methods for colon mapping and brain mapping, respectively. It is well known [21] that the harmonic function u is the minimizer of the Dirichlet functional

$$F(u) = \frac{1}{2}\int\int_{\Sigma} |\nabla u|^2 dS$$

$$u|_{\partial\sigma_0} \equiv 0,\, u|_{\partial\sigma_1} \equiv 1. \tag{1}$$

Let $PL(\Sigma)$ denote the finite dimensional space of piecewise linear functions on Σ. We seek a minimizer u of $F(u)$ in the space $PL(\Sigma)$. Using linearity and integration by parts, it is straightforward to show [23] that the minimization of $F(u)$ reduces to the solution of a system of linear equations, one equation for each vertex V in $\Sigma \backslash (\sigma_0 \cup \sigma_1)$:

$$\sum_{W \in \Sigma \backslash (\sigma_0 \cup \sigma_1)} D_{VW} \, u_W = - \sum_{W \in \sigma_1} D_{VW}. \tag{2}$$

Here, the sparse symmetric matrix $D = (D_{VW})$ is zero except when $V = W$ or V and W are connected by an edge in the triangulation. In the latter case, we suppose VW is an edge belonging to two triangles VWX and VWY. A computation [22] shows that D_{VW} is then given by the formula

$$D_{VW} = -\frac{1}{2}(\cot \angle X + \cot \angle Y). \tag{3}$$

The diagonal elements of D may be computed from

$$D_{VV} = - \sum_{W \neq V} D_{VW}, \tag{4}$$

from which it can be seen that D positive semi-definite.

In short, u may be found by solving the sparse linear system (2) using standard methods from numerical linear algebra. We have found that the solution of this system can be found in under 5 minutes on a single processor Sun Ultra 10, for a surface consisting of $100,000$ triangles. Once the solution u is found, the center points may be found simply by dividing up the interval $[0, 1]$ into a number of sub-intervals, and calculating for each sub-interval the center of mass of the vertices with corresponding values of u.

5 Discussion

Our tool has been used in two clinical studies. The first study was a virtual colonoscopy of a 66 year old female patient. The original source for the colon surface was an axial helical CT scan. The patient had two real polyps and we added phantom polyps in the data set that ranged in size from 4.9 to 7.2 mm. The 3D model of the colon was color coded by an algorithm used in Haker *et al* [3], and similar to [24] . Centerline points through the colon were extracted automatically with the algorithm described in Section 4 and the Virtual Endoscopic tool created a fly-through path for the camera. All Polyps can be clearly detected during the fly-through. A movie of a virtual colonoscopy can be seen at http://www.ai.mit.edu/projects/medical-vision/virtual-endoscopy/.

Our system was also used in a cardiovascular project to observe branching vessels from the inside of the heart atrium of a 34 year old male patient. The original source for the atrium and vessels was an MRI scan. In this study, it was important to place the endoscope at precise contiguous positions in the atrium to observe views into the vessels.

For further evaluation, we plan to ask physicians to compare and evaluate the different features of our program and create fly-throughs of various anatomic structures. We are also working on developing and integrating a suite of automatic path generation algorithms into the Virtual Endoscopy module. Our goal is to produce centerlines automatically through surface models of any topology. This will allow the users to choose to bring automation into the path generation process. We would also like to add the ability to have branching paths with the option for the user to make only a portion of the path active for the fly-through.

Acknowledgements This work has been supported by NIH grants P41 RR13218-01, P01 CA67165-03, R01 RR11747-01A and NSF grant ERC 9731748. We would like to thank Lauren O'Donnell, Carl-Fredrik Westin, Peter Everett and Dave Gering for their useful suggestions and ideas throughout the development phase. Finally we would like to thank Polina Golland and Lilla Zollei for their helpful comments on drafts of this paper.

References

[1] D. Gering, A. Nabavi, R. Kikinis, N. Hata, L. O'Donnell, E. Grimson, F. Jolesz, P. Black and W. Wells. An integrated visualization system for surgical planning and guidance using image fusion and an open mr. *J. Magn. Reson. Imaging*, 13:967–975, 2001.

[2] Mayo Clinic. Biomedical image resource. *http://www.mayo.edu/bir*.

[3] S. Haker, S. Angenent, A. Tannenbaum, and R. Kikinis. Nondistorting flattening maps and the 3d visualization of colon ct images. IEEE Trans. on Medical Imaging, 19:pp. 665–670., 2000.

[4] A. Schreyer, J. Fielding, S. Warfield, J. Lee, K. Loughlin, H. Dumanli, F. Jolesz, R. Kikinis. Virtual cystoscopy: Color mapping of bladder wall thickness. *Invest Radiol*, 35(5):331–334, 2000.

[5] Claussen Seemann. Hybrid 3d visualization of the chest and virtual endoscopy of the tracheobronchial system: Possibilities and limitations of clinical applications. *Lung Cancer*, 32(3):237–46, 2001.

[6] U. Ecke, L. Klimek, W. Muller, R. Ziegler, W. Mann. Virtual reality: Preparation and execution of sinus surgery. *Comp Aid Surg*, 3:45–50, 1998.

[7] P. Rogalla, M. Werner-Rustner, A. Huitema, A. van Est, N. Meiri, B. Hamm. Virtual endoscopy of the small bowel: Phantom study and preliminary clinical results. *European Radiology*, 8:563–567, 1998.

[8] D. Vining, S. Aquino. Virtual bronchoscopy. *Clin Chest Med*, 20(4):725–30, 1999.

[9] C. Kay, D. Kulling, R. Hawes, J. Young,P. Cotton. Virtual endoscopy - comparison with colonoscopy in the detection of space-occupying lesions of the colon. *Endoscopy*, 32(3):226–32, 2000.

[10] R. Summers, P. Choyke, N. Patronas, E. Tucker, B. Wise, M. Busse, H. Jr Brewer, R. Shamburek. Mr virtual angioscopy of thoracic aortic atherosclerosis in homozygous familial hypercholesterolemia. *J Comput Assist Tomogr*, 25(3):371–7, 2001.

[11] Freeflight software. *http://www.vec.wfubmc.edu/software/*.

[12] The Multidimensional Image Processing Lab at Penn State University. *http://cobb.ece.psu.edu/projects/quicksee/quicksee.htm*.

[13] M. Wan, F. Dachille, and A. Kaufman. Distance-field based skeletons for virtual navigation. *Visualization 2001, San Diego, CA*.

[14] F. Jolesz, W. Lorensen, H. Shinmoto, H. Atsumi, S. Nakajima, P. Kavanaugh, P. Saiviroonporn, S. Seltzer, S. Silverman, M. Phillips, R. Kikinis. Interactive virtual endoscopy. *AJR*, 169:1229–1235, 1997.

[15] T. Nakagohri, F. Jolesz, S. Okuda, T. Asano, T. Kenmochi, O. Kainuma, Y. Tokoro, H. Aoyama, W. Lorensen, R. Kikinis. Virtual pancreatoscopy. *Comp Aid Surg*, 3:264–268, 1998.

[16] M. Fried, V. Moharir, H. Shinmoto, A. Alyassin, W. Lorensen, L. Hsu, R. Kikinis. Virtual laryngoscopy. *Annals of Otology, Rhinology and Laryngology*, 108(3):221–226, 1998.

[17] The 3D Slicer. *http://www.slicer.org*.

[18] Hearn and Baker. *Computer Graphics*. Prentice Hall, New Jersey, 1997.

[19] W. Schroeder, H. Martin, and B. Lorensen. The visualization toolkit. Prentice-Hall, New Jersey, 1996.

[20] A. Tannenbaum L. Zhu, S. Haker. Conformal flattening maps for the visualization of vessels. *Submitted for Publication*, 2001.

[21] J. Rauch. Partial differential equations. Springer-Verlag, New York, 1991.

[22] T. Hughes. The finite element method. Prentice-Hall, New Jersey, 1987.

[23] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis. On the laplace-beltrami operator and brain surface flattening. IEEE Trans. on Medical Imaging, 18:pp. 700–711., 1999.

[24] R. Summers, C. Beaulieu, L. Pusanik, J. Malley, R. Jr Jeffrey, D. Glazer, S. Napel. An automated polyp detector for ct colonography - feasibility study. *Radiology*, 216:284–90, 2000.