3D Virtual Colonoscopy with Real-time Volume Rendering

Ming Wan[1], Wei Li[1], Kevin Kreeger[1], Ingmar Bitter[1], Arie Kaufman[1]
Zhengrong Liang[2], Dongqing Chen[2] and Mark Wax[3]

Center for Visual Computing (CVC)
and Departments of Computer Science and Radiology
State University of New York at Stony Brook
Stony Brook, NY 11794-4400

Abstract

In our previous work, we developed a virtual colonoscopy system on a high-end 16-processor SGI Challenge with an expensive hardware graphics accelerator. The goal of this work is to port the system to a low cost PC in order to increase its availability for mass screening. Recently, Mitsubishi Electric has developed a volume-rendering PC board, called VolumePro, which includes 128 MB of RAM and vg500 rendering chip. The vg500 chip, based on Cube-4 technology, can render a 256^3 volume at 30 frames per second. High image quality of volume rendering inside the colon is guaranteed by the full lighting model and 3D interpolation supported by the vg500 chip. However, the VolumePro board is lacking some features required by our interactive colon navigation. First, VolumePro currently does not support perspective projection which is paramount for interior colon navigation. Second, the patient colon data is usually much larger than 256^3 and cannot be rendered in real-time. In this paper, we present our solutions to these problems, including simulated perspective projection and axis aligned boxing techniques, and demonstrate the high performance of our virtual colonoscopy system on low cost PCs.

Keywords: Virtual colonoscopy, virtual endoscopy, 3D medical imaging, scientific visualization, volume rendering, interactive navigation, cancer diagnosis, hardware acceleration, perspective projection, virtual reality.

1 Introduction

Colorectal carcinoma is the third most commonly diagnosed cancer and the second leading cause of death from cancer in the United States. Often it is diagnosed at an advanced stage, after the patient has developed symptoms, explaining its high mortality rate [1]. Since most cancers arise from polyps over a 5 to 15 year period of malignant transformation, screening programs to detect small polyps less than 1 cm in diameter have been advocated. Optical colonoscopy is the commonly used diagnostic procedure. Unfortunately most people do not follow this recommendation because of the associated risk, discomfort, and high cost. In order to dramatically increase the number of people willing to participate in screening programs, efforts have been made towards a computer-based screening modality, called 3D virtual colonoscopy, as an alternative to optical colonoscopy, by employing advanced computer graphics and visualization techniques [2, 3, 4].

The virtual colonoscopy system takes a spiral CT scan of the patient's abdomen after the entire colon is fully cleansed and distended with room air or CO_2. Several hundred high resolution CT images are rapidly acquired during a single breathhold of about 30-40 seconds, forming a volumetric abdomen data set. A model of the real colon is then segmented from the abdomen data set. It can be viewed either by automatic planned navigation following the center-line of the lumen of the colon, providing a general

[1]Email:{mwan,liwei,kkreeger,ingmar,ari}@cs.sunysb.edu
[2]Email:{jzl,dchen}@clio.rad.sunysb.edu
[3]Email:mwax@radiol.som.sunysb.edu

overview of the inner colonic surface, or by interactive navigation for a more detailed study of suspicious regions. In our previous work, we developed such a virtual colonoscopy system on high end 16-processor SGI Power Challenge with expensive Infinite Reality graphics hardware acceleration [4, 5, 6, 7]. We have already confirmed that we can visualize polyps as small as 3 mm, and polyps that have been detected in optical colonoscopy have also been identified with our virtual colonoscopy. Our work in this paper is to make our technology easily accessible on a low cost PC (Personal Computer) so as to increase the availability as a mass screening procedure.

Real-time rendering rates of 10-30 frames per second are critical for an interactive virtual navigation and detection inside the virtual human colon. We previously proposed and implemented two different real-time colonic rendering techniques on high-end SGI workstations, respectively using fast surface rendering [4, 5] and direct volume rendering techniques[6, 7]. Fast surface rendering time was achieved by exploiting the Infinite Reality graphics hardware accelerator on SGI workstations to display those colon surfaces visible to the current view. The colon surface was extracted from the colon volumetric data in a preprocessing stage. Fast direct volume rendering rates were reached by using 16 processors to directly render the colon images from the original 3D volumetric colon data. Our preliminary experimental results indicated that the direct volume rendering technique provided more realistic colonic images, flexible visualization of interior structures for polyps and other abnormalities, and shorter preprocessing time. Physicians have confirmed that our direct volume rendering images of the human colon are very close to what they observed in optical colonoscopy. Consequently, when our virtual colonoscopy system is ported and implemented on the low-end PC, it is critical to maintain the high image quality and rendering speed of our direct volume rendering techniques.

Mitsubishi Electric has developed a volume-rendering PC board, which supports real-time direct volume rendering rates, called VolumePro. Currently, the first generation VolumePro board from Mitsubishi [8, 9], which is based on the Cube-4 architecture developed at SUNY at Stony Brook [10], contains a vg500 chip and 128 MB of RAM, allowing an entire colon dataset to fit on one rendering board. We ported the existing navigation software to Microsoft Windows NT on a PC and utilized the VolumePro board to perform preliminary quality analysis of the volume rendering system. Note that the vg500 chip can render 500 million samples per second in parallel projection. If one sample is taken per voxel (either 1 byte or 2 bytes) so that no data is missed, then the vg500 chip can render a 256^3 volume at 30 frames per second. Evidently, the current board with vg500 chip cannot render the entire large colon dataset at sufficient frame rates.

Our new PC version of virtual colonoscopy has been developed by using Microsoft Visual C++, based on MFC. Our simulation results indicated that the quality of the images produced by the VolumePro board was very close to those volume rendering images generated on the SGI workstations. This is because the VolumePro board supports a full lighting model and complete 3D interpolation. It is worthy to point out that direct usage of the VolumePro board does not meet our interactive colon navigation requirements. First, VolumePro currently does not support perspective projection which is paramount for interior colon navigation. Second, the patient colon data is usually much larger than 256^3 (e.g., 512^3) and cannot be rendered in real-time. In this paper, we present our solutions to these problems, including simulated perspective projection in Section 2 and axis aligned boxing techniques in Section 3, and demonstrate the high performance of our virtual colonoscopy system on low-cost PCs in Section 4.

2 Uniform Error Perspective Projection

As mentioned earlier, the current version of the VolumePro board does not support perspective projection, whereas it is a critical requirement for virtual colonoscopy. We simulated the process on the board by dividing the volume into multiple image-plane-aligned slabs. Each slab is rendered in parallel mode. Then, the resulting images are used as textures which are mapped onto hexagons appropriately placed in front the camera. Finally, all the hexagons are alpha-blended from back to front.

A key issue in this method is how to chose the number of slabs and the thickness of each slab. The naive method is to evenly slice the dataset along the viewing direction resulting in a uniform thickness for each slab. The number of slabs can either be fixed or variable, depending on the trade-off between speed and quality. Generally speaking, the larger the number of slabs, the less error will be introduced by this method.

In perspective projection, objects at a distance have a smaller 2D projection on the image plane than objects closer to the eye. Therefore, it is natural to employ an adaptive method[11], where thinner slabs are used in the front and thicker slabs in the back. We develop the following formula to select the thickness of slabs and relate them to an error measure.

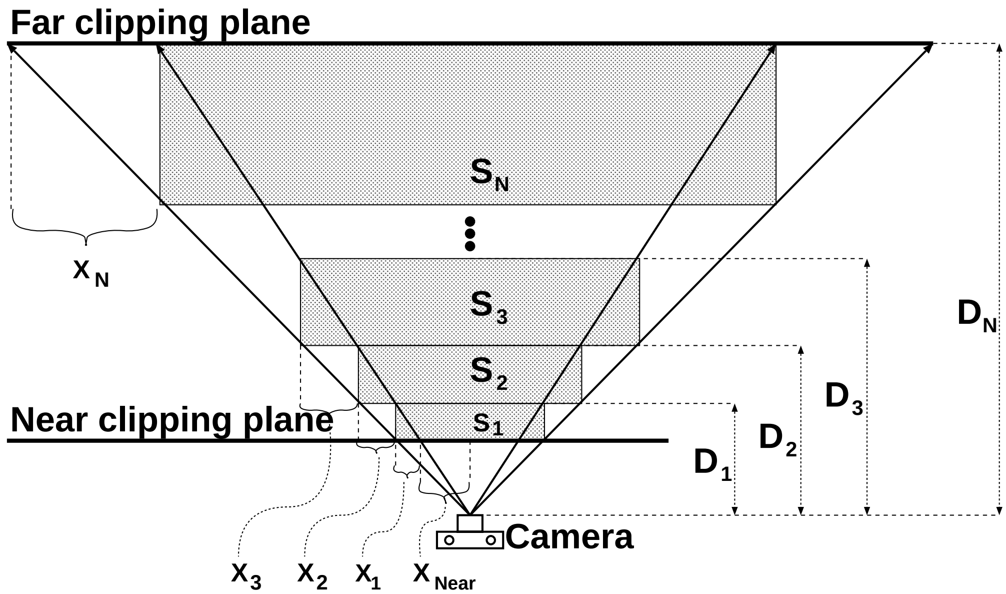

Figure 1: Determination of optimal slab numbers and thickness for simulated perspective projection.

Seen in figure 1, the largest distortion along the X axis between two adjacent slabs S_i and $S_{i+1}$is X_{i+1}, which can be used as an error measure for the distortion along the X axis. For each adjacent slab pairs S_i and S_{i+1}, the error is proportional to the thickness of slab i. It is apparent that such an error along Y should be the proportional to the one along X. Our principle is to make all the projections on the image plane of X_i be equal to each other, that is, make the distortion uniformly distributed among all slabs. Therefore, we define the error measurement as follows:

$\varepsilon =$ the magnitude of the projection of the maximal distortion X_i of adjacent slab pairs in X direction and $X_i = X_j$ for all $1 \leq i, j \leq N$ where N is the total number of slabs.

Denote the distance from the view point to the apparent near and far plane as $near$ and far, respectively. Without losing generality, suppose $near = 1$. D_i is the distance from the view point to the near face of the ith slab. The thickness of the ith slab is $D_{i+1} - D_i$. Then, we have:

$$\begin{cases} \sum_{i=1}^{n} X_i + X_{near} = D_n X_{near} \\ \frac{X_i}{\varepsilon} = \frac{D_i}{1} \end{cases}$$

Where X_{near} equals half of the width of the window on the near plane.

$$\begin{cases} X_i = D_i * \varepsilon \\ \sum_{i=1}^{n} D_i \varepsilon = (D_n - 1) X_{near} \end{cases}$$

$$D_n \varepsilon + \sum_{i=1}^{n-1} D_i \varepsilon = D_n X_{near} - X_{near}$$

$$D_n (X_{near} - \varepsilon) = \sum_{i=1}^{n} D_i \varepsilon + X_{near}$$

Hence

$$D_{n+1} - D_n = \frac{1}{X_{near} - \varepsilon} D_n \varepsilon$$

$$\begin{cases} D_{n+1} = \frac{X_{near}}{X_{near} - \varepsilon} D_n \\ D_1 = \frac{X_{near}}{X_{near} - \varepsilon} \end{cases}$$

The solution of this recursion can be expressed explicitly as:

$$D_n = (\frac{X_{near}}{X_{near} - \varepsilon})^n \tag{1}$$

This means, $D_i, i = 1..N$, is a geometric series.
Assume the total number of slabs is N,then

$$Far \leq D_N = (\frac{X_{near}}{X_{near}-\varepsilon})^N$$

Clearly, we should choose $Far = D_N$. For a given error measure ε ,the number of slabs should be:

$$N = \left\lceil \ln Far / \ln(\frac{X_{near}}{X_{near}-\varepsilon}) \right\rceil$$

After the number of slabs is determined, we can calculate the new error bound ε' as:

$$\varepsilon' = X_{near}(1 - \frac{1}{\sqrt[N]{Far}})$$

where

$$\varepsilon' \leq \varepsilon$$

Finally, the thickness of each slab can be derived by the recursion in equation (1). With this method, our system can automatically determine the smallest number of slabs required by the given error bound, and optimally (in the sense of the error measurement defined in this paper) choose the thickness of slabs. These parameters are actually determined by the angle-of-view and the depth of the back plane far. Far is data-dependent and can be determined by a visibility detection algorithm.

3 Rendering with Subvolumes

Direct usage of the Uniform Error Perspective Projection with the full 512^3 colon dataset does not provide interactive navigation due to the multiple passes required. Fortunately, a good feature of colon navigation is that the camera is located inside the colon most of the time, and thus a large percentage of the data falls outside of the view-volume. Moreover, we often chose a transfer function so that the colon wall is opaque. Therefore, when the camera is flying inside the colon, only a small portion of the data is visible. Consequently, we can divide the dataset into subvolumes and only render the visible portions. Furthermore, parts of the dataset which do not contain the colon will never be loaded onto the board memory.

In perspective projection, it is difficult to composite images of two subvolumes that share a face which is not parallel to the image plane. Therefore, in our system, any subvolume to be rendered for a certain image can have only shared faces that are parallel to the image plane.

Owing to the above reasons, we divide the volume dataset uniformly along each axes. Each subvolume has a 50% overlap with its nearest neighbors in all three directions. The overlapping is because we want to make sure each rendering taken by VolumePro board is carried out on a single subvolume rather than requiring the rendition to cross boundaries.

During navigation, the best subvolume is determined according to the position, viewing direction and angle of view of the camera. Considering the subvolumes as a three dimensional array, we calculate the indices of the best subvolume in the three dimensions separately since the indices for the different dimension are irrelevant. That is, for each index, we imagine the subvolumes are just one dimensional.

Assume the size of dataset is N voxels along a certain axis X; the size of subvolumes along the same axis is M. Then, the dataset is divided into $2\frac{N}{M} - 1$ subvolumes along X. For a given set of camera parameters, we first find the subvolume sv whose center is closest to the x coordinate of the camera. Then we decide whether the view frustum is completely on one side of the line crossing the center of sv and is orthogonal to axis X. If not, the best subvolume is sv. Otherwise, choose subvolume $sv + 1$ or $sv - 1$ accordingly. For example, in Figure 2, camera geometry 1 will utilize subvolume S_v, camera geometry 2 will use subvolume S_{v-1} and camera geometry 3 will use subvolume S_{v+1}.

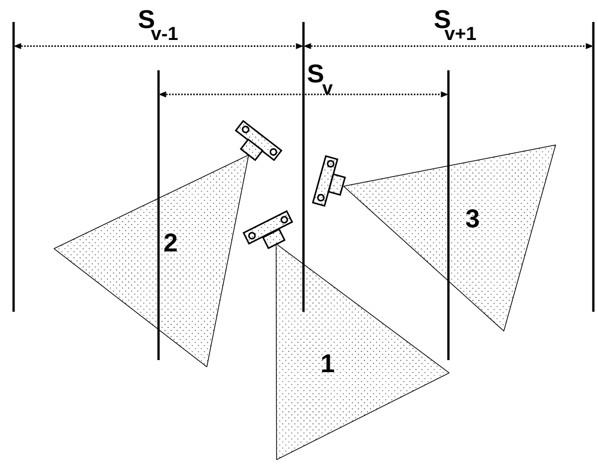

Figure 2: Determination of the best-fitting subvolume for the current view.

The most significant disadvantage of this method is the duplication of data. This method requires up to 8 times the storage due to the 50% overlap. Therefore, the current version of VolumePro board can only handle volume data up to 256^3.

Furthermore, the subvolume-rendering can be combined with our perspective projection method, since both of them need to divide the original dataset. The time that the VolumePro board needs to render is almost proportional to the size of the subvolume being rendered. As a result, the fewer the voxels contained in subvolumes, the faster the VolumePro board can render.

In our perspective rendering method, each slab is rendered separately. Then, for each slab, we render the smallest possible volume, which is the smallest axis-aligned bounding box of the intersection of the view-volume, current slab and current visible box.

4 Experimental Results

Figure 3 is a screen shot of the virtual colonoscopy user interface. The center panel is the navigation window in the endoscopic view. The left panel provides an outside overview of the colon and the right panel shows the corresponding sagittal, coronal and transverse slices.

Figure 4 and Figure 5 display the differences between choosing slab thicknesses using our uniform error method and the naive method of uniform slab thickness. In Figure 4, both images are generated with 14 slabs, however the right image shows artifacts (in the white box) where the soft tissue behind the colon wall is visible in the image. This artifact occurs more frequently in uniform sampling because of the higher error measure ε for some slabs over others. Similarly, Figure 5 shows the increased error for the uniform thickness method with 16 slabs.

5 Conclusions

In order to increase the availability of our 3D virtual colonoscopy system for mass screening, we implemented it on a low-cost PC by using the first generation real-time volume rendering board, called VolumePro. We proposed both adaptive perspective projection simulation and colon volume subdivision strategies to support real-time navigation inside large patient colon data sets with natural internal perspective views. The high quality images of volume rendering are guaranteed by the full lighting model and complete 3D interpolation supported by the VolumePro board. As a result of our work, a volume rendering based virtual colonoscopy system with real-time performance can be widely available to medical practitioners on low-cost PC.

6 Acknowledgments

This work has been supported by grants from NIH #CA79180, the Center for Biotechnology, and E-Z-EM Inc. The pipe and patients' data sets were provided by the University Hospital of the State University of New York at Stony Brook. Special thanks to Sarang Lakare, Mie Sato, and Qingyu Tang for their contribution to the virtual colonoscopy project. Thank Qi Ke for the help of analysis of the recursive formula of the adaptive thickness of slabs.

References

[1] J. Mandel, J. Bond, J. Church, and D. Snover (1993). "Reducing Mortality from Colon Cancer Control Study". *New England J Med 328*, 1993, 1365-1371.

[2] W. Lorensen, F. Jolesz, and R. Kikinis (1995). "The Exploration of Cross-Sectional Data with a Virtual Endoscope". In R. Satava and K. Morgan (eds.), *Interactive Technology and the New Medical Paradigm for Health Care*, IOS Press, Washington D. C., 1995, 221-230.

[3] D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin (1994). "Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality". Presented at *the Annual Meeting of the American Roentgen Ray Society*, New Orleans, April, 1994.

[4] L. Hong, A. Kaufman, Y. Wei, A. Viswambharn, M. Wax, and Z. Liang (1995). "3D Virtual Colonoscopy". *Proc. 1995 Symposium on Biomedical Visualization*, 1995, 26-32.

[5] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He (1997). "Virtual Voyage: Interactive Navigation in the Human Colon". *Proc. SIGGRAPH '97*, August 1997, 27-34.

[6] S. You, L. Hong, M. Wan, K. Junyaprasert, A. Kaufman, S. Muraki, Y. Zhou, M. Wax, and Z. Liang (1997). "Interactive Volume Rendering for Virtual Colonoscopy". *Proc. IEEE Visualization '97*, Phoenix, AZ, October 1997, 433-436.

[7] M. Wan, Q. Tang, A. Kaufman, Z. Liang, and M. Wax (1999). "Volume Rendering Based Interactive Navigation within the Human Colon". *Proc. IEEE Visualization' 99*, 1999, 397-400.

[8] R. Osborne, H. Pfister, H. Lauer, N. McKenzie, S. Gibson, W. Hiatt, and T. Ohkami (1997). EM-Cube: An Architecture for Low-Cost Real-Time Volume Rendering. *1997 SIGGRAPH / Eurographics Workshop on Graphics Hardware*, 131–138, ACM Press, August 1997.

[9] H. Pfister, J. hardenbergh, J. Knittel, H. Lauer, L. Seiler (1999). "The VolumePro Real-Time Ray-Casting System". *Proc. SIGGRAPH '99*, August 1999, 251-260.

[10] H. Pfister and A. Kaufman (1996). "Cube-4: A scalable architecture for real-time volume rendering". *Proc. IEEE Symposium on Volume Visualization' 96*, 47-54, ACM Press, October 1996 .

[11] K.A. Kreeger, I. Bitter, F. Dachille, B. Chen, A. Kaufman. "Adaptive Perspective Ray Casting". *Symposium on Volume Visualization.* Research Triangle Park, NC. October 19-20, 1998.