

Perspective Virtual Endoscopy with VolumePro Parallel Rendering

Kevin Kreeger, Li Wei, Sarang Lakare and Arie Kaufman[†]

Center for Visual Computing (CVC)
and Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11749-4400

Abstract

Virtual Endoscopy, a non-invasive examination of hollow tubular structures within a human body, has the potential to revolutionize the field of medical diagnostics, therapy and surgical planning. It is well known that, compared to polygonal approximations to iso-surfaces, Volume Rendering techniques provide more accurate images. Additionally, volume rendering provides more flexible visualization, by allowing more structure than just the single iso-value to be examined. Unfortunately, previous volume rendering techniques were either too slow, too low quality, or required too expensive of a machine to be widely utilized by medical practitioners. In this paper we describe our method to produce images with perspective divergence using the parallel projection on the Mitsubishi/Real-Time Visualization VolumePro board developed at SUNY Stony Brook.

Keywords: Volume Rendering, Perspective Projection, Virtual Endoscopy, Rendering Hardware

1 Introduction

For the past 6 years we have been developing a system to provide non-invasive, affordable, comfortable mass screening for colorectal cancer. Approximately 150,000 new cases of colorectal cancer are diagnosed every year. Unfortunately, it is most often discovered after the patient has developed symptoms. Therefore, the American Cancer Society has recommended a colon Exam every 3 years after the age of 50 to detect polyps which may lead to cancer. Currently, optical colonoscopy and barium enema are the only two procedures available for examining the entire colon to detect polyps larger than 5mm in diameter, which are clinically considered to have a high probability of being malignant. In optical colonoscopy, a fiber optical probe is introduced into the colon through the rectum. By manipulating the tiny camera attached to the tip of the probe, the physician examines the inner surface of the colon to identify abnormalities. This invasive procedure requires intravenous sedation, takes about one hour, and is expensive. Barium enema requires a great deal of physical cooperation from the patient when the X-ray radiographs of the colon are taken at different views. Additionally, its sensitivity can be as low as 78% in detecting polyps in the range of 5mm to 20mm [5]. Of even greater concern, is that most patients do not follow their physicians advice to undergo such a procedure because of the associated risk, discomfort, and high cost. Consequently, a new massive screening technique which is accurate, cost-effective, non-invasive and comfortable would be extremely valuable. It would have the potential for a large population colon screening and could detect small colon polyps at their early stages. The detection and removal of these small polyps can completely cure the patients condition.

The technique we have been developing, *virtual colonoscopy* [1], is an alternative procedure for examining the entire colon [1, 4, 7, 8]. In general, this procedure consists of three steps. First, the

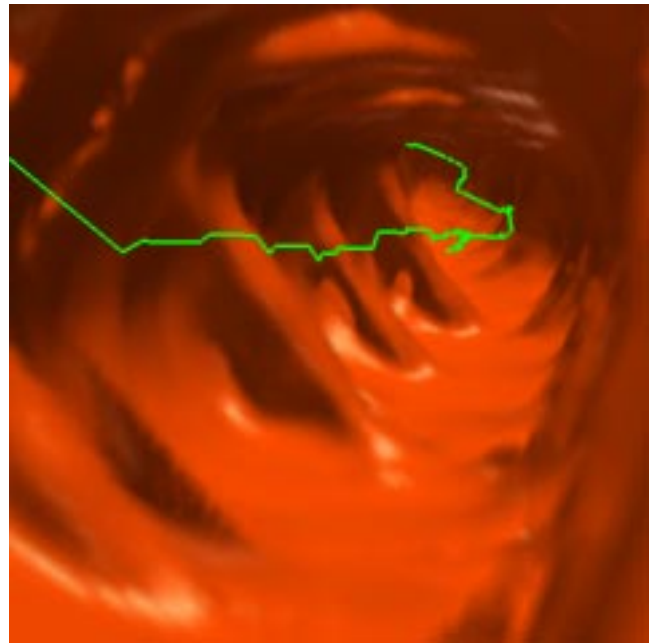


Figure 1: Example endoscopic view of a patient's colon with a perspective projection.

patient's colon is cleansed (either physically, or electronically [3]) and inflated with air in a way similar to that of optical colonoscopy. Second, while the patient is holding his or her breath, a helical CT scan of the patient's abdomen is taken, capturing a sequence of 2D slices which covers the entire range of the colon. This scan takes 30 – 45 seconds and produces several hundred transaxial slices of 512×512 pixels, which are subsequently reconstructed into a 3D volume of 100 – 250 MBytes. It can then be viewed either by automatic planned navigation or interactive navigation. This non-

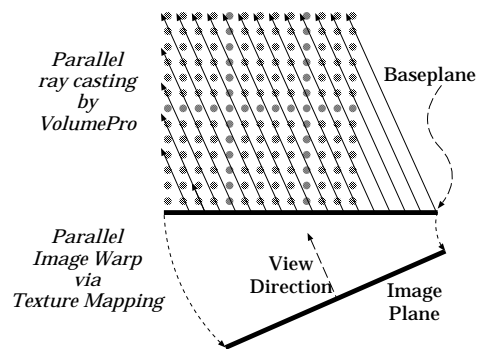


Figure 2: Two step volume rendering with the VolumePro board by taking advantage of 3D graphics cards to perform image warp.

[†]{kkreeger,liwei,lsarang,ari}@cs.sunysb.edu

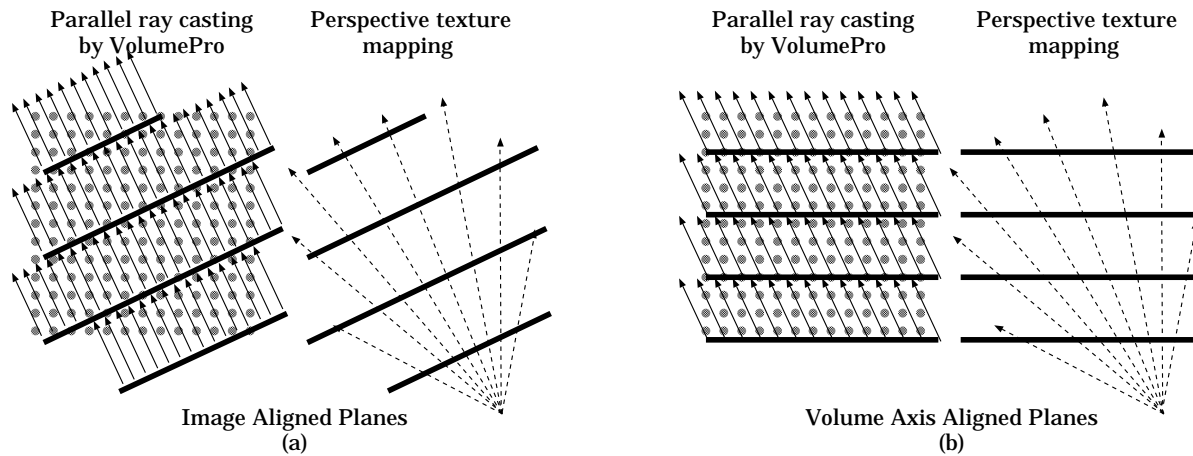


Figure 3: Multipass perspective approximation using slabs aligned (a) parallel to the image plane or (b) orthogonal to a volume axis.

invasive procedure can potentially improve the diagnostic sensitivity and specificity with fewer complications, lower expense, shorter examination time, and reduced patient discomfort [8].

Over the course of our development of this technique, we have utilized high end SGI workstations to achieve the required performance of interactive rendering speeds. Unfortunately, the use of expensive machines precludes the use of the system as a mass screening device since only a few medical practitioners would be able to afford it and health care providers would probably not cover the charges. Therefore, a low cost PC-based solution is desirable. Recently, Mitsubishi/Real-Time Visualization has released the VolumePro PC board which achieves 30Hz rendering of 256^3 volume datasets [6]. We have, therefore, begun using this board to perform virtual colonoscopy in a PC platform making the procedure affordable to medical practitioners everywhere and cost-efficient to health-care providers such as HMO's and Insurance companies. The image in Figure 1 was produced on a \$5000 PC (including the cost of the VolumePro board).

While, we have so far targeted virtual colonoscopy, various other endoscopic procedures are possible using our methods to view tubular organs in the human body. For example, Bronchial tubes, the esophagus, arteries, stomach etc can all be examined endoscopically on a low cost PC with our methods.

The remainder of the paper is organized as follows. In Section 2 we discuss the VolumePro Board and introduce two limitations that we had to solve to allow endoscopic perspective rendering. In Section 3 we show how we achieve perspective divergence from parallel projections. In Section 5 we present our method to perform visible voxel culling for endoscopic viewing. In Section 4 we present the performance and quality tradeoffs, and end with conclusions in Section 7.

2 VolumePro Hardware Rendering Board

VolumePro is the the world's first real-time volume rendering accelerator for consumer PCs [6]. The ray casting algorithm is implemented in a slice-order method. Since, gradient estimation, classification, and per-sample Phong Illumination are computed in hardware, high quality images are guaranteed. The VolumePro ray casting algorithm with data access based on the shear warp factorization of the view matrix, but full 3D interpolation and sample placement to eliminate the 45 degree artifacts associated with the original shear-warp algorithm [2]. In this design, the rays are cast through the volume (in parallel along the view direction) onto a *baseplane* which is subsequently warped onto the final image. This process, shown in 2D in Figure 2 takes advantage of the texture mapping

on conventional 3D graphics cards to perform the final warp into an image. The VolumePro board works most efficiently with AGP graphics cards rather than PCI graphics cards. Since the VolumePro board sits in a PCI slot, this allows the baseplane image to be sent to the 3D graphics card more quickly.

Performing volume rendering on a PC with the Volume Pro hardware has advantages over existing methods. Since hardware acceleration is used, it is much faster than software methods. Since the board computed per sample illumination, the images are higher quality than 3D texture map based solutions. Unfortunately, the VolumePro board does not quite meet the stringent requirements for perspective endoscopic views of medical datasets. Specifically, the board is not capable of perspective projections, rather only parallel. Obviously, perspective projections are a requirement of endoscopic views, parallel projections of "tubes" result in "donuts" being displayed in the final image. Additionally, medical CT datasets – typically $512 \times 512 \times (100-400)$ – are too large for the board to render in real time. Luckily, in endoscopic views of tubular structures, large portions of the dataset are obscured. We utilize this feature to cull the dataset down to a size which the board can handle.

3 Perspective Divergence with Parallel Projections

To enable the use of the VolumePro board to create endoscopic images of human organs, we devised a method to create the impression of perspective divergence using a multipass algorithm. Rather than creating one baseplane image with the VolumePro board and then texture mapping that single image, our approach renders multiple baseplane images. Subsequently, we composite these images together by texturing them onto planes placed in 3D space at their respective slab positions and utilize a perspective projection on the 3D graphics card. Each of the multiple baseplane images represents a "thick slab" of the volume data and are created with parallel projection along the view direction. The slabs can be aligned either parallel to the final image plane, or orthogonal to one of the three volume axes. Both approaches are shown in Figure 3.

Obviously, this does not produce an exact perspective projection, as the samples taken along the parallel "sub-rays" are not at the same position as samples that would be taken if true perspective rays were cast through the volume. This is shown for both image aligned slabs and volume aligned slabs in Figure 4. Note that the amount of error depends only on the angle of the specific ray with respect to the view direction and is the same for either alignment method. For example, the rays in the center of the image are cast very close to the view direction and all sub rays for that one ray

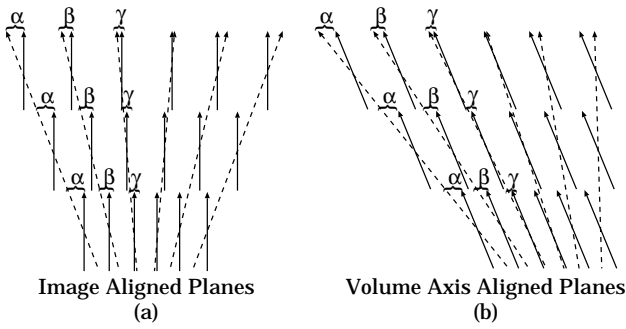


Figure 4: Ray Sample position error for (a) image aligned planes and (b) volume axis aligned planes

have an error γ that is small. The rays towards the edges of the image, however, have a larger error α , although it is also similar for all sub rays along the ray. For this reason we call this a perspective approximation algorithm. The amount that this error is visible depends upon the dataset and the transfer function. For example, if there are high frequency opaque edges, the noticeable error is disturbing, while translucent “gel” views of mostly homogenous data appear quite nice.

For endoscopic views, we normally wish to see the “lumen” walls of the object we are “flying” through. For example, in virtual colonoscopy, we search for polyps which normally appear as bumps in the colon wall. Therefore, we wish to optimize the image quality for this case. Figure 5 shows the artifacts which can occur in endoscopic views of organ walls. In Figure 5A we see that normal perspective ray casting causes all rays to hit the colon wall and, assuming the transfer function makes is opaque, displays the wall, and not the soft tissue behind it. However, with the perspective approximation, the parallel rays in (b₁) create the baseplane images in (b₂). Since we map the slab images to planes that are placed in their 3D position in space, and then perform a 3D projection, the slab image texels are projected to the image pixels such that the soft tissue behind the wall show past sometimes (see rays in b₃). Furthermore, the viewer is normally focusing on the colon wall to the edges of the images, not on the center of the image, which shows portions of the colon further from the viewpoint as is shown in Figure 6. Since features in these far away portions are smaller, the user typically focuses on the closer walls (although not the extreme edges of the image) and waits until flying further down the tube to view the areas which are currently in the distance (and, hence, towards the center of the image where the artifacts are reduced).

To minimize the overhead when rendering the multiple slab images. The VolumePro board allows setting *ActiveSubvolumes* at axis-aligned 32 voxel boundaries. This feature causes only the voxels within the defined cuboid to be processed. However, the image-

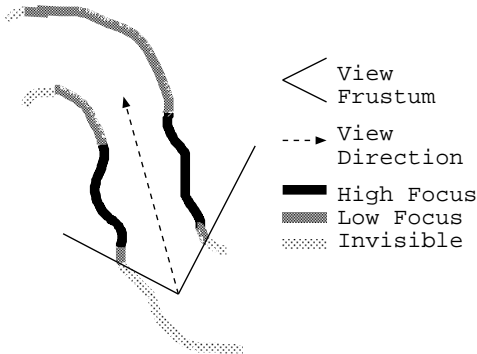


Figure 6: Portion of a colon dataset showing area where user typically focuses attention.

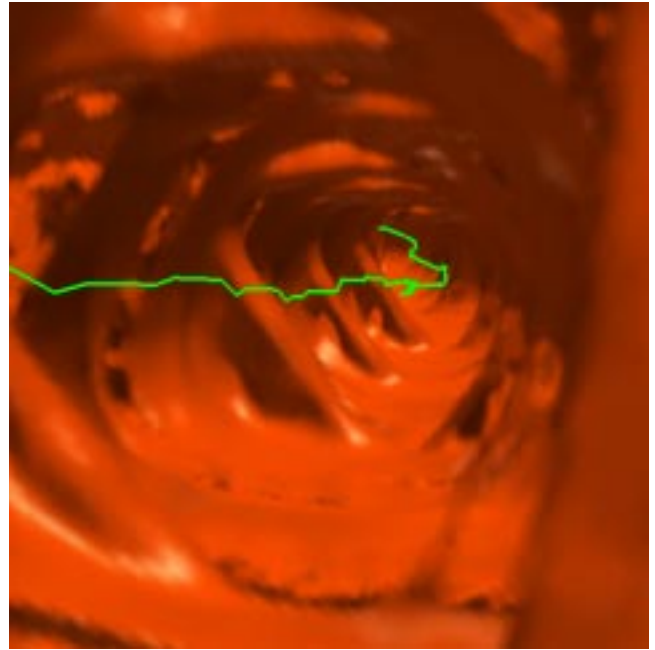


Figure 7: Example image with few slabs showing artifacts from the large distance between baseplane images.

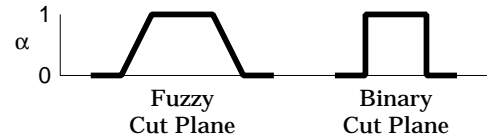


Figure 8: Falfoff creates a fuzzy cutplane transition versus a binary in/out decision which causes aliasing.

aligned method shown in Figure 3A requires slabs which are not axis aligned. For this reason, we set the active subvolume to the multiple-32 bounding box around the current slab and then use the VolumePro *thick cut plane* to define the actual portion of the volume which will contribute to the image. Using the the thick cut plane allows arbitrary orientation, location and thickness of the slab. Utilizing the thick cut plane feature of VolumePro has the benefit that aliasing can be reduced by setting a *falfoff* on the edges of the cut-plane. This feature specifies a width to the transition between full and zero opacity creating a fuzzy cutplane instead of binary (see Figure 8).

In fact, for the Volume axis aligned method in Figure 3B, we also use thick cut planes, because the slabs are normally smaller than 32 voxels thick.

Previously we showed that, rays diverge as an exponential function of the distance from the camera. We used this fact to adapt the number of rays to remain close to the underlying voxel density, by defining regions that were twice as thick as the previous region towards the camera. We utilize the same method here to define the thickness of the slabs to minimize the artifacts in the image.

4 Rendering Performance/Quality Analysis

It should be noted that the thickness of the slabs is directly proportional to the amount of error in the sample positions. Since thinner slabs create a larger number of slabs to cover the volume, this results in an image quality / rendering time tradeoff that must

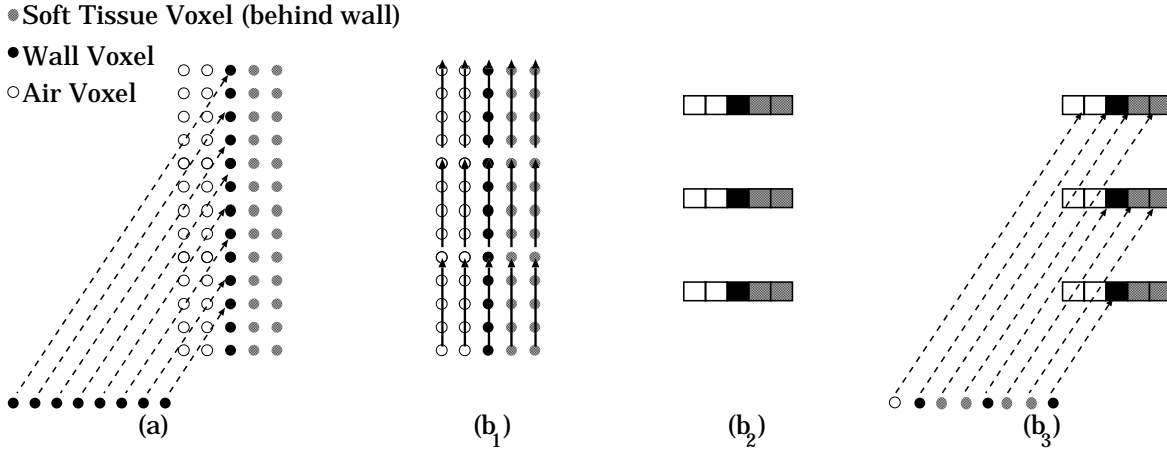


Figure 5: Endoscopic artifacts due to perspective approximation. (a) all rays in true perspective hit wall. (b) The parallel rays (b_1) create multiple baseplane images (b_2) which allow the perspective rays to see “behind” the wall (b_3)

1. Find nearest and furthest Z for given volume and view frustum
2. $cutplanewidth = \frac{furthestZ - nearestZ}{cutplaneCount}$
3. Load $cutplanewidth$ on VolumePro card
4. Compute cutplane equation A, B & C
5. $D = furthestZ$
6. load A, B, C, D on VolumePro card
7. Call VolumePro render with primary baseplane buffer
8. **while** $D < nearestZ$
9. load $A, B, C, (D - cutplanewidth)$ on VolumePro card
10. Call VolumePro render with opposite baseplane buffer
11. Fetch primary baseplane buffer
12. Set Texture with primary baseplane buffer
13. Transform baseplane by $cutplanewidth$
14. Render baseplane (performs warping)
15. toggle baseplane buffers
16. $D = D - cutplanewidth$
17. **return**

Algorithm 1: Pseudo-code to perform perspective approximation algorithm overlapping VolumePro and Transfer/Texture/Blend on 3D graphics card by using two baseplane buffers.

be optimized for each application. For each slab that is rendered, the VolumePro board runs through the voxels and creates a baseplane. Each baseplane image must be transferred from the VolumePro board to the 3D graphics card for texturing. Thus the bandwidth used is relative to the number of slabs. Also, each image is texture mapped and blended into the frame buffer. We performed our testing on a PC equipped with a Diamond Viper v770 3D graphics card (which contains the nVidia TNT chip). We noticed that enabling the OpenGL *AlphaTest* greatly increased the rendering speed due to the fact that many pixels in the baseplane textures were completely transparent. We utilize *GLTexSubImage* to replace each baseplane texture image instead of *GLTexImage*. This also made a noticeable difference in the rendering speed, implying that texture object setup on this card takes a significant amount of time compared with texture data replacement. Another feature that gave good performance was disabling *GLDepthTest*. This resulted in 0.01 seconds faster per frame. Interestingly, we hypothesized that much performance was lost on performing the blend calculations for each fragment, but we tested with blend turned off (of course the image was incor-

rect), and the rendering time was exactly the same as with blend turned on.

To study the effect of the multipass algorithm on the different

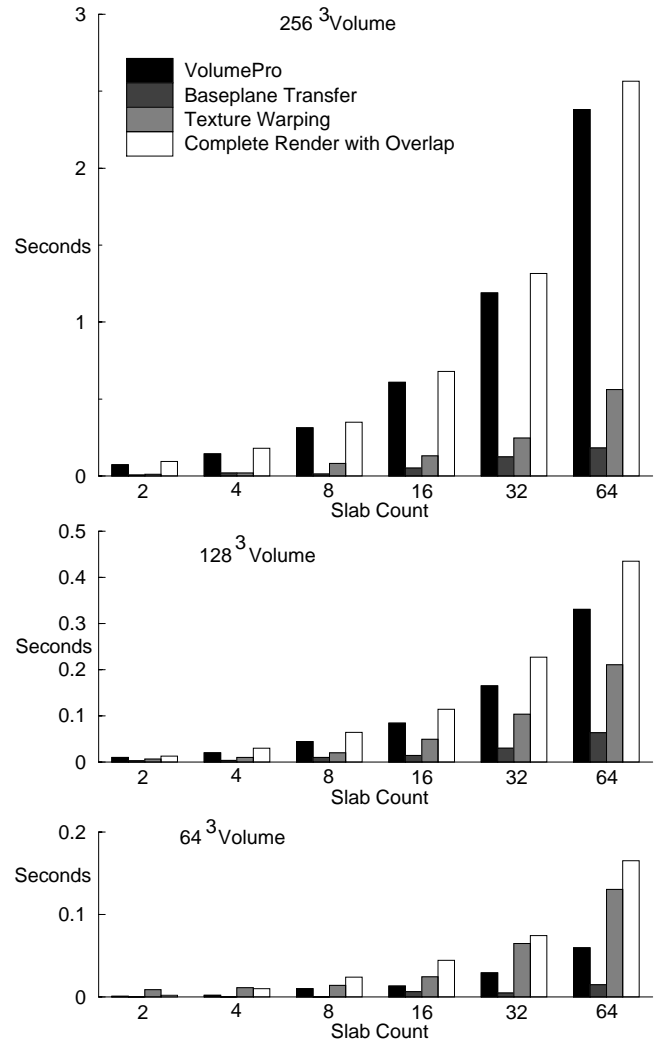


Figure 9: Rendering time versus slab count for different sized volumes.

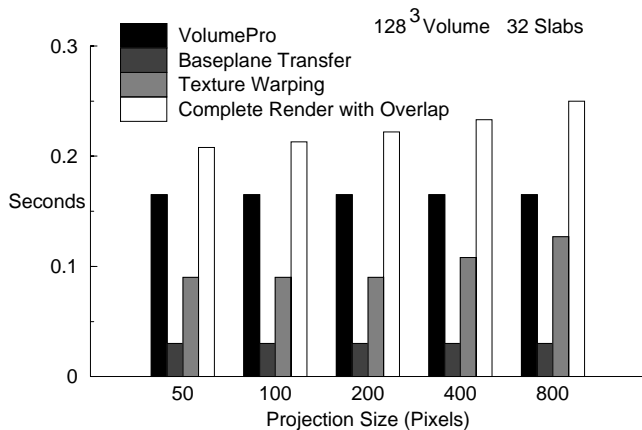


Figure 10: Rendering time versus zoom factor.

portions of the system we timed the system. The graphs in Figure 9 show the time to perform different portions of the algorithm for different volume sizes and different slab counts. We measured the time spent by the VolumePro board, the time spent transferring the baseplane texture maps to the 3D graphics card, and the time spent by the 3D graphics card rendering the baseplane images. Additionally, we measured the frame time for the complete rendering when the VolumePro board processing was overlapped with the rest of the processing, notice that while there is some overhead, the total time is less than the sum of the portions showing that some overlapping can occur. The three graphs in Figure 9 come from rendering 64^3 , 128^3 , and 256^3 volume respectively. Notice that the larger the volume, the more the total runtime is dominated by the VolumePro rendering time, and that the smaller the volume, the more the 3D baseplane texture warping operation dominates the frame time. This occurs, because the texture warping depends on the image size, which didn't change in these experiments, while the VolumePro board performs work proportional to the volume size. In Figures 10 and 11 we kept the same size volume, and changed the image size. In Figure 10 we zoomed the image up. Notice that for all projection sizes, the VolumePro and baseplane transfer times are equal, while the texture warping times increase. Notice that the overlapping rendering times also increase. In Figure 11, we kept the object scale constant, but rotated the volume around the $+y$ axis and $+x$ axis. This creates larger baseplane projections of the volume data (see [6] for a more detailed explanation of how and why this occurs). Thus the VolumePro rendering times increase a bit, while the transfer times increase as well as the projection times. This occurs because, although the scale is constant, the projection of the cuboid shaped volume at different angles covers more pixels on the screen, causing more frame buffer accesses.

5 Volume Culling

The VolumePro board is capable of 500 million samples per second. This is sufficient to render a 256^3 volume at 30Hz, however, you may choose to utilize the 500 million samples in other ways. For example, a $256 \times 256 \times 128$ volume can be rendered with 2 times Z-supersampling also at 30Hz, or a $512 \times 512 \times 256$ dataset can be rendered at 7.5 Hz, any order which adds up to 500 million samples per second. Of course there is also frame setup, baseplane image transfer, and image warping overheads that must be considered (although the warp of one frame can be overlapped with the ray casting of the next frame to lower some of the overhead). Typical virtual colonoscopy datasets are $512 \times 512 \times 400$. Rendering this dataset alone would result in only 5 frames per second. While this

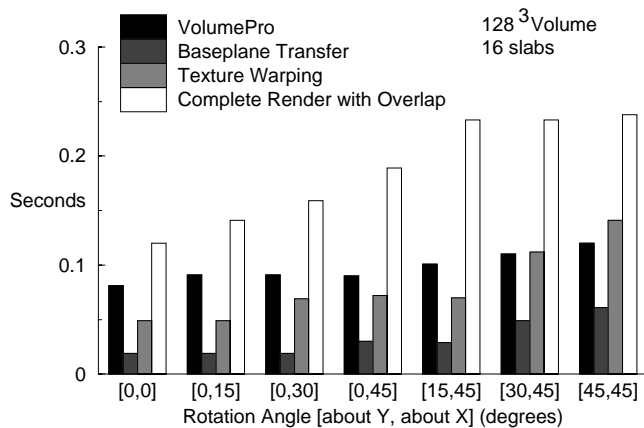


Figure 11: Rendering time versus view angle.

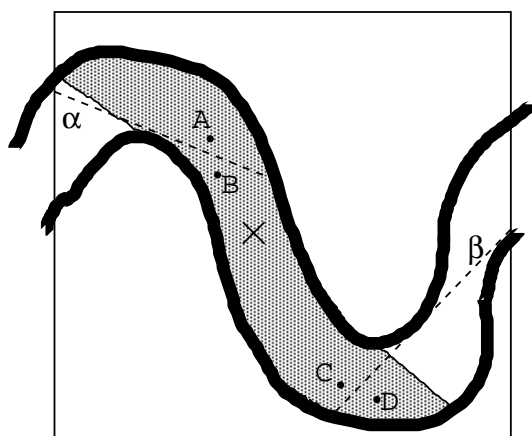


Figure 12: Multiple-32 aligned bounding box for point \times also works for points B and C.

is close to interactive frame rates, the multipass perspective algorithm lowers this below the 1 frame per second mark and would result in an unusable system. Luckily, virtual endoscopy, and particularly virtual colonoscopy, is creating images of twisted tubular structures. For this reason, from any point within the organ, large portions of the dataset are non-visible. We use this feature to cull off these portions and render only a small subset of the volume which we guarantee will contain all the visible data from the current viewpoint.

Since the caching and queuing on the VolumePro board requires cuboid shaped volumes of multiple-32 boundaries, we cre-

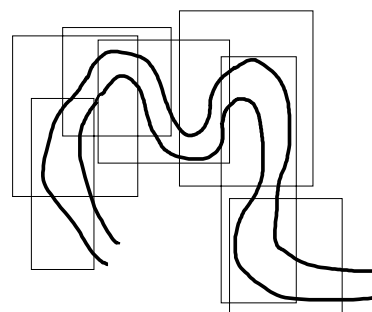


Figure 13: overlapping boxes.



Figure 14: External overview of the colon used in the tests.

ate a cuboid like this of all visible voxels for each viewpoint. One method for this would be to preprocess the volume and store, for each possible camera position the visible subvolume. Then when rendering, just lookup this subvolume and utilize it. For colonoscopy, the camera can move anywhere within the colon dataset. If we store subvolumes for any voxel within the colon, we can take the current camera location and lookup the “nearest” voxel and use its subvolume. Unfortunately, there are 200,000 (RIGHT INGMAR/MIE) voxels within a typical colon. This would create a large table to lookup the correct subvolume. Additionally, due to the topology of the colon, many voxels (especially those close to each other) will utilize the exact same subvolume. Consider the case shown in Figure 12. Due to the twisted nature of the colon, only the shaded portion is visible from point \times . However, when a multiple-32 bounding box is placed around this region, the box shown extends past the boundaries of the shaded region. For this reason, points which lie below line α (e.g. point B) will be able to only see voxels contained in the same multiple-32 subvolume, while point above line α (e.g. point A) will be outside the multiple-32 subvolume and will require a different one. Likewise at the other point where the colon exits the subvolume, line β separates points C and D as the cutoff point where the visibility changes. You can see the large area between the two dashed lines that will utilize the same multiple-32 subvolume. When switching from one subvolume to another, there will always be overlap (see Figure 13).

We create the subvolumes and regions in a pre-processing step by walking through the colon from one end to the other and using a greedy algorithm. We begin by creating the first subvolume which includes all visible voxels from the end of the colon. We then walk through the colon searching for the first point where we can “see” outside the current multiple-32 subvolume. At this point we create a new one which encompasses all the currently visible voxels, then walk again. This is repeated the entire way through the colon. While this does not create the minimum number of subvolumes, the number created is easily manageable. For example, for the colon in Figure 14, 49 subvolumes were created with this method. The size of the subvolume rendered is drastically reduced, with

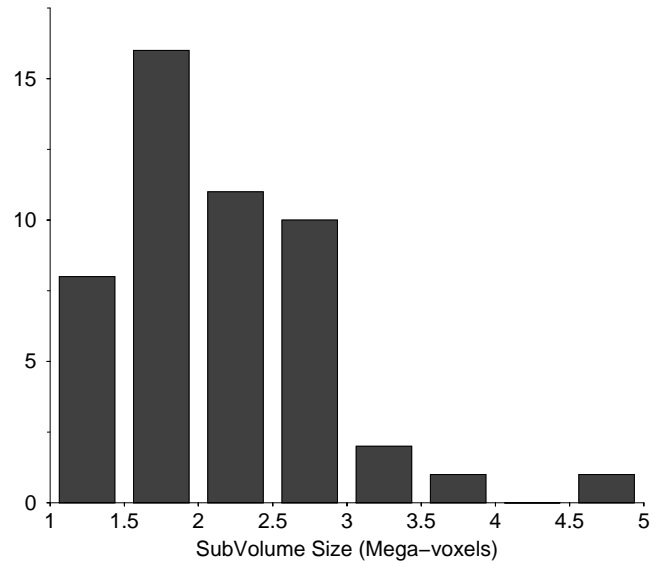


Figure 15: Histogram of number of subvolumes within the range of voxel sizes.

the average being two mega-voxels, or the same order of magnitude as 128^3 . Of course the subvolumes are not all exact cubic, for example, some are $160 \times 192 \times 96$ or $160 \times 128 \times 128$ or $128 \times 96 \times 96$. The histogram of number of subvolumes per size is shown in Figure 15. Notice that most are on the order of magnitude of two mega voxels, with some smaller and a very few extra large. The extra large ones occur at the long stright colon sections where curves don’t cause occlusion to occur for a long way.

THIS WE ACTUALLY DID BY HAND, IT PROBABLY WON’T BE DONE FOR REAL BEFORE THE SUBMISSION FOR REVIEW DATE, BUT HOPEFULLY SHOULD BE DONE BY THE FINAL COPY SUBMISSION DATE. EVERYTHING I WROTE HERE APPLYS TO BOTH HOW IT WAS DONE BY HAND AND (HOPEFULLY) HOW IT WILL BE DONE AUTOMATICALLY. SHOULD THIS SECTION BE CHANGED AT ALL DUE TO THIS?

6 System Performance

We have utilized the perspective approximation and subvolume culling algorithms for a PC version of our virtual colonoscopy system. Our system provides interactive navigation which responds to user input, so we cannot pre-compute fly-through frames for real-time playback, but rather must achieve interactive frame rates. In previous papers [?] we have presented our navigation algorithms as well as methods for interactive rendering with surfaces on high end graphics workstations and with volume rendering on expensive parallel workstations.

To implement our virtual colonoscopy system on a PC, we utilize the first generation of the VolumePro board in a PC with 512MBytes memory, a 500Mhz PIII processor and a Diamond v770 Ultra AGP graphics card.

For the perspective approximation with the parallel VolumePro rendering, we utilize 20 slabs per frame with 1 voxel of falloff on the edges for anti-aliasing. We utilized the image-plane aligned slabs since they resulted in fewer image artifacts. We utilize the subvolume culling algorithm and achieved between 4 and 8 frames per second rendering rates depending on the size of the subvolume being displayed. The image quality is higher than achievable with alternative solutions such as 3D texture mapping. Figure 16 shows images from a virtual navigation within a patients colon. (a) is the

view down the pipe, (b) is a view about to turn a bend, (c) shows a 3mm polyp (about the minimal resolution where polyps are visible), (d) shows an 8 mm polyp as the virtual camera flies past while (e) shows the same polyp after navigating the camera to view the structure head-on. Finally, (f) shows a "virtual biopsy" of the same 8 mm polyp performed by simply adjusting the transfer function so that all densities are semi-translucent and the color ranges from blue, through green to red. This frame shows the structure and dense core of the polyp that is normally hidden by the colon wall.

7 Conclusions

The methods presented in this paper allow virtual colonoscopy on a PC class machine enabling physicians and clinics everywhere affordable access to this life saving technology.

While the rendering is between 4-8 frames per second, we would like to sustain over 12 frames per second. Future work to speed up the rendering includes more advanced culling algorithms instead of the greedy approach currently employed. We would also like to attempt the use of multiple VolumePro boards rendering slabs in parallel since, even with overlapping VolumePro calculations with 3d Graphics processing, the runtime is still dependant on the VolumePro processing time.

The second generation of the VolumePro board, which is due out late in 2000, will allow significant speedup in this algorithm since it provides arbitrary start and stop points for rays at much better than multiple-32 slice boundaries. We predict a 4-5 times speedup by removing the overhead of processing large number of voxels that don't contribute to each individual slice image. With this we hope to allow supersampling and to be able to increase the number of slabs.

References

- [1] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang. 3D Virtual Colonoscopy. In *IEEE Symposium on Biomedical Visualization*, pages 26–32, Oct. 1995.
- [2] P. Lacroute and M. Levoy. Fast Volume Rendering using a Shear-warp Factorization of the Viewing Transform. In *Computer Graphics, SIGGRAPH 94*, pages 451–457, July 1994.
- [3] S. Lakare, M. Wan, M. Sato, and A. Kaufman. Segmentation Rays for Electronic Cleansing in Virtual Colonoscopy. In *Submitted to EuroGraphics*, 2000.
- [4] W. Lorensen, F. Jolesz, and R. Kikinis. The Exploration of Cross-Sectional Data with a Virtual Endoscope. *Interactive Technology and New Medical Paradigm for Health Care*, pages 221–230, 1995.
- [5] C. Morosi, G. Ballardini, and P. Pisani. Diagnostic Accuracy of the Double-Contrast Enema for Colonic Polyps in Patients with or without Diverticular Disease. *Gastrointest Radiology*, 16:346–347, 1991.
- [6] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The VolumePro Real-Time Ray-Casting System. In *Computer Graphics, SIGGRAPH 99*, pages 251–260, Aug. 1999.
- [7] G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffrey, and S. Napel. Perspective Volume Rendering of CT and MR Images: Applications for Endoscopic Imaging. *Radiology*, 199:321–330, May 1996.
- [8] D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin. Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality. *Annual Meeting of American Roentgen Ray Society*, page 104, 1994.

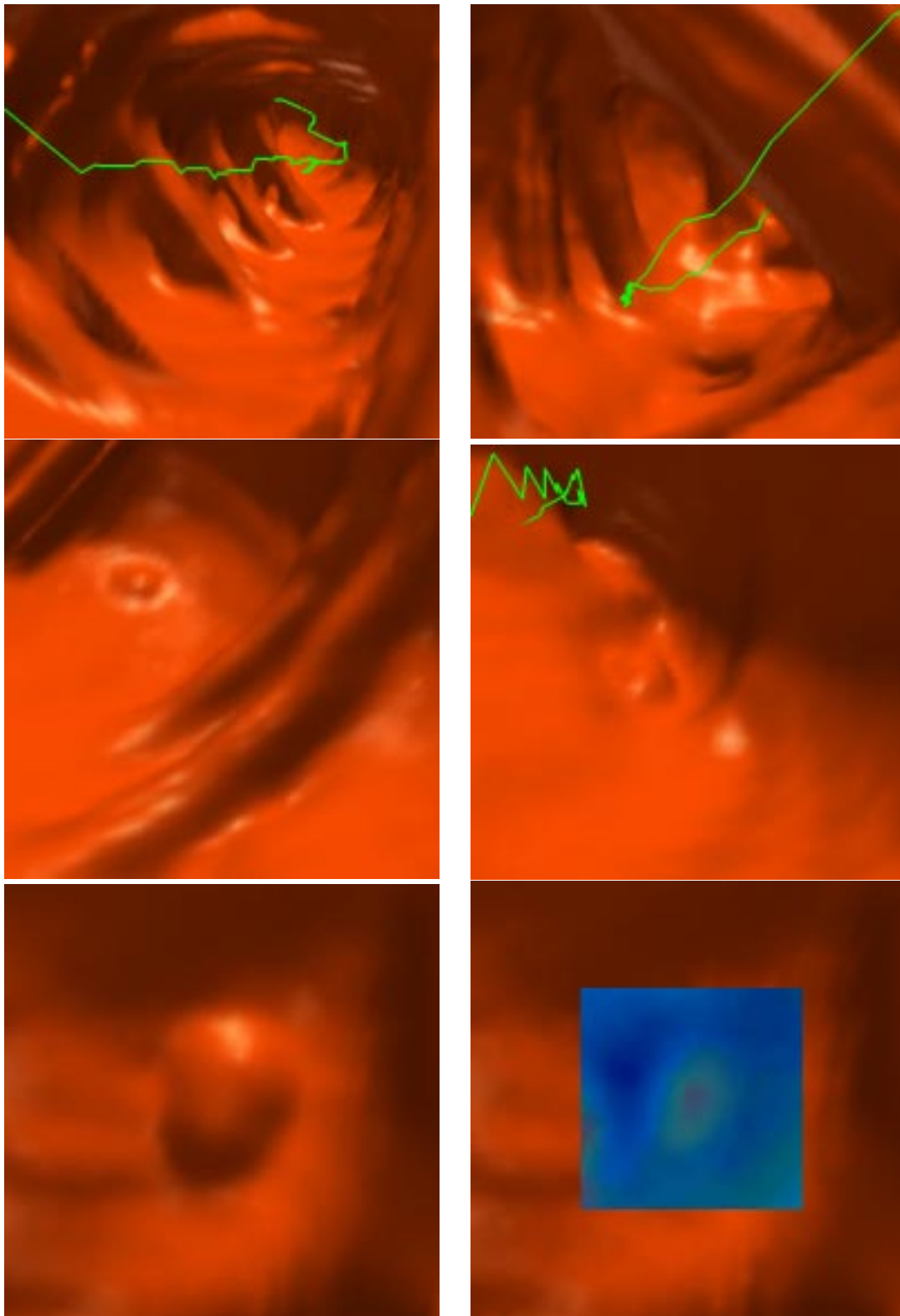


Figure 16: *Scenes from a virtual colonoscopy procedure: (a) view down a long stright portion of the colon, (b) view going around a bend in the colon, (c) view close to a small 3 mm polyp, (d) view passing an 8mm polyp, (e) looking stright on the the same 8mm polyp, (f) virtual biopsy of the 8mm polyp.*