3D Digital Cleansing Using Segmentation Rays

Sarang Lakare, Ming Wan, Mie Sato, and Arie Kaufman[†]

Center for Visual Computing (CVC)
and Department of Computer Science
State University of New York at Stony Brook
Stony Brook, NY 11790-4400

Abstract

We propose a novel approach for segmentation and digital cleansing of endoscopic organs. Our method can be used for a variety of segmentation needs with little or no modification. It aims at fulfilling the dual and often conflicting requirements of a fast and accurate segmentation and also eliminates the undesirable partial volume effect which contemporary approaches cannot. For segmentation and digital cleansing, we use the peculiar characteristics exhibited by the intersection of any two distinct-intensity regions. To detect these intersections we cast rays through the volume, which we call the segmentation rays as they assist in the segmentation. We then associate a certain task of reconstruction and classification with each intersection the ray detects. We further use volumetric contrast enhancement to reconstruct surface lost by segmentation (if any), which aids in improving the quality of the volume rendering.

Keywords: Volume Segmentation, Segmentation Rays, Partial Volume Voxels, Volume Rendering, Virtual Endoscopy, Virtual Colonoscopy

1 Introduction

Over the last decade, volume rendering techniques have grown rapidly, and currently are able to generate accurate results at interactive frame rates. As a result of this, a lot of research today focuses on using these rendering techniques for virtual screening of organs in the human body [7]. Often these organs have complex structures which require careful segmentation before they are fit to be screened. Thus, for accurate diagnosis of a patient, segmentation plays a very important role. In addition, a speedy diagnosis could be crucial for these virtual techniques to be able to replace the contemporary techniques some day.

Segmentation needs for the virtual screening techniques could range from simple thresholding to careful removal of unwanted material. Segmentation could also get complicated due to partial volume effect, which can cause unwanted and non-existing surfaces to pop up during rendering. With our method, we eliminate this partial volume effect by detecting and eliminating all partial volume voxels. Our approach is a flexible one, and can be used for a variety of segmentation needs with little or no modification. It combines the contemporary thresholding and flood-fill techniques with two new techniques, namely, segmentation rays and volumetric contrast enhancement.

We explain our segmentation algorithm using an example application of an upcoming virtual screening technique, Virtual Colonoscopy [3][4][9], which is being developed at the State University of New York (SUNY) at Stony Brook. We now briefly outline the segmentation requirements of the virtual colonoscopy system and previous solutions before explaning our solution using our new method.

2 The Experiment

The aim of virtual colonoscopy is to detect potentially cancerous growths in the colon, called polyps, by letting the doctor virtually fly through the colon. This eliminates the need for the painful and highly uncomfortable optical colonoscopy exam that the patient would otherwise have to go through. Early detection of polyps is important because their removal, before they get malignant, can completely cure the patient [8]. Polyps with a diameter of more than 5 mm are considered potentially malignant and need to be removed. Thus, the virtual colonoscopy aims at detecting the smallest of polyps.

One problem with all current colonoscopy techniques is that they require a clean colon lumen for accurate detection of polyps. Residual material inside the colon could be falsely interpreted as part of the colon in virtual colonoscopy, and in optical colonoscopy, it could hinder the movement of the camera and/or the doctor's view of the colon, with the chance that some polyps might remain undetected. Thus, as preparation for the colonoscopy, the patient undergoes a physical colon cleansing. This includes either washing the colon with large amounts of liquids or administrating medications and enemas to induce bowel movements. This colon preparation is often more uncomfortable and unpleasant than the colonoscopy examination itself.

To make the virtual colonoscopy system even more friendly to the patient, it is necessary to bypass the physical cleansing of the colon. Thus, there is a need for segmenting the residual material out of the colon, giving a clean colon to the rendering algorithm. As a first step to accomplish this segmentation, a new bowel preparation scheme was developed at SUNY Stony Brook [5][10]. This helps segmentation by enhancing the stool and fluid (the residual material in the colon) densities.

A CT scan of the patient's abdomen is then taken and reconstructed into a 3D dataset. The dataset obtained is much more complex than the one obtained with a pre-cleansed colon. The complexity arises because of the large amount of fluid and stool residing inside the colon (Figure 1). Although these unwanted residues have been enhanced, they do not have a clear boundary due to partial volume effect. The situation is even worsened by the finite resolution and low contrast of the CT scanner. We now briefly summerise the shortcomings of contemporary segmentation algorithms for this problem.

Segmentation by thresholding

The simplest segmentation approach is thresholding. The human

[†] {lsarang, mwan, mie, ari}@cs.sunysb.edu

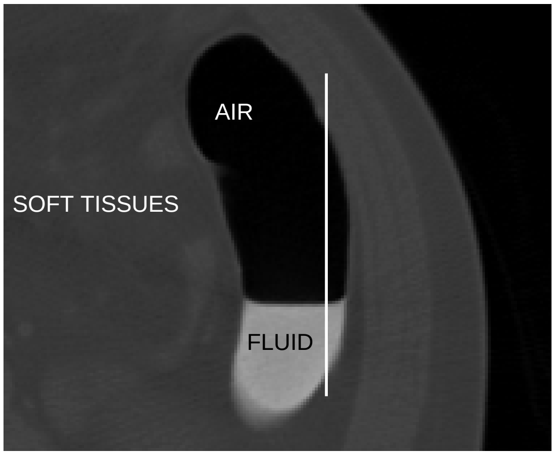

Figure 1: *Part of a colon as seen from a traverse slice.*

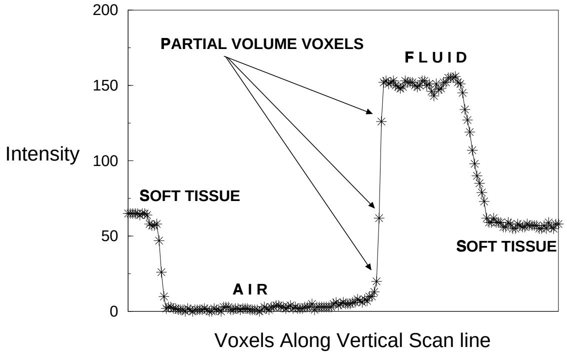

Figure 2: *Intensity profile at the boundary of air and fluid.*

abdomen consists mainly of three distinct density regions: air, soft-tissue, and high density materials (which include the bone). A CT scan assigns different intensities to these materials and we classify them based on these intensities. For a physically cleansed colon, thresholding would be enough to accurately segment the colon for rendering in virtual colonoscopy [3][4]. However, with the new bowel preparation scheme, bones are not the only high density material present in the abdomen. Residual fluid and stool, which are enhanced with barium, also have CT image intensities similar to those of bone, thus complicating the segmentation. Although thresholding gives the fastest results, it has many disadvantages which we list below.

First, thresholding does not remove partial volume voxels. Figure 2 shows the intensity profile along a vertical line from top to bottom as shown in Figure 1. We observe that at the boundary of two regions with different intensities lie voxels, whose intensities do not match either of the two regions. We name these voxels the *Partial Volume Effect (PVE)* voxels. These voxels are undesirable since they are incorrectly classified when thresholding is used. For example, in Figure 2, the voxels between the fluid and the air lie in the soft-tissue range. Hence, they are marked as soft-tissue voxels and would not be removed. The adverse effect on segmentation is immediately evident and is shown in Figure 3. Although the high density fluid/stool has been removed, a thin soft-tissue-like boundary still exists, which is not present in reality.

Second, the thresholds for each range of intensities are very sensitive. A slight change to these thresholds could lead to a change in the outcome of the segmentation, especially the contour of the inner surface of the colon.

Third, thresholding also gives rise to aliasing effects at the inner

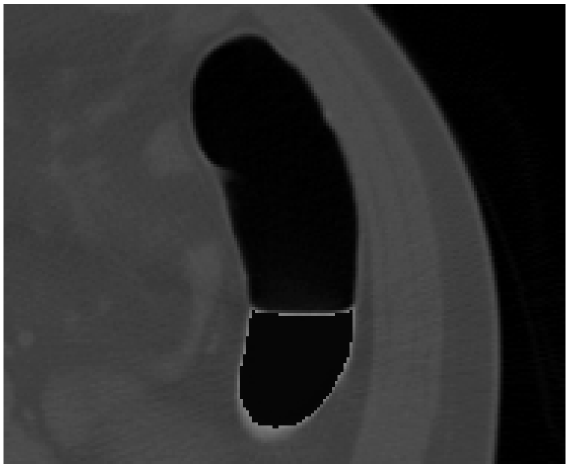

Figure 3: *Result of segmentation by thresholding.*

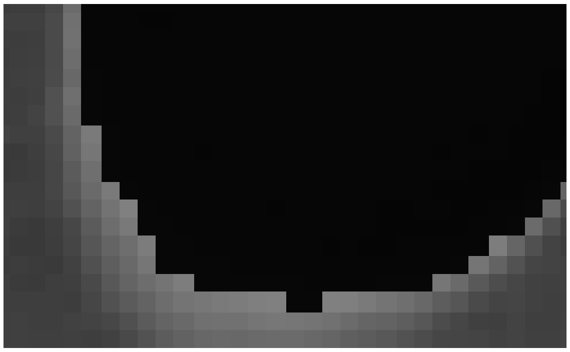

Figure 4: *Aliasing effects of segmentation by thresholding.*

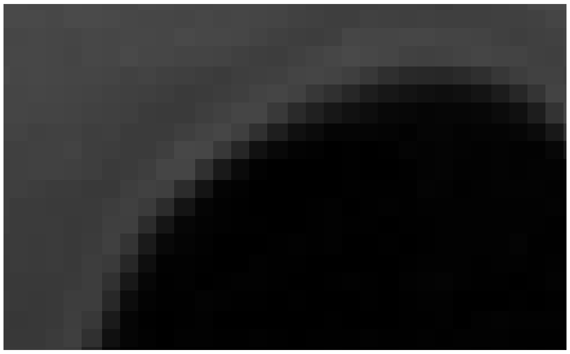

Figure 5: *A normal colonic mucosal surface without fluid or stool attached to it.*

colon boundary. It is immediately evident when we take a closer look at the segmented volume (Figure 4). The intensity values move sharply from soft-tissue range to air-range. This is undesirable from the point of view of the volume rendering. A sharp boundary also means the absence of colonic mucosa (a thin soft-tissue layer) that is present on the inner surface of the colon. Colonic mucosa is key to the detection of the polyps and hence its removal is undesirable. For comparison, we also show an example interior surface of the mucosa of the colon with no residual stool attached (Figure 5).

Segmentation by morphological operations

A succession of morphological operations, such as dilates and erodes, could also be used for segmentation [2]. For example, we could perform a flood-fill on all the fluid and stool regions, and then use a sequence of dilates and erodes to remove the *PVE* voxels. However it is well known that dilates followed by erodes can fill in holes, and erodes followed by dilates can remove noise [2]. Thus, they could affect the inner contour of the colon, as it is highly twisted.

This method would also require seed points to be placed in each and every region filled with fluid or stool for performing the flood-fill. This could be a cumbersome task considering the large number of such regions. In addition, it may require a lot of human intervention. This could also slow down the entire process of segmentation and may result in some fluid/stool regions being neglected.

Other segmentation approaches

Recently, Liang et al. [6][10] reported three approaches to colon segmentation. The first two are very slow (3 hours and 30 minutes respectively on a Pentium II), which limit their clinical application. The third approach, which takes 6 minutes on a PC platform, generates feature vectors for each voxel and then applies thresholding. Thus, the result may be dependent on strict thresholds, which is undesirable. Their approach also removes some colon mucosa voxels which may result in the incorrect detection of polyps. Wyatt et al. [11] described another fully automatic segmentation, but their method does not aim at accuracy and is very slow (60-65 minutes on an SGI onyx).

We now outline our algorithm for segmentation and digital cleansing. We then descibe at length how we use it to successfully tackle the problems we described above for digital cleansing of the colon and how it can be applied to any situation with similar demands for segmentation.

3 Our Algorithm

The crux of our algorithm is based on the unique characteristic property at the interesection of two distinct-intensity regions. This unique characteristic is the intensity profile as we move in a direction approximately normal to the intersecting surfaces.

The main constituents of our algorithm are the use of segmentation rays. These are named so because they assist in segmentation. When these rays traverse through the volume, they compare their intensity profile with some pre-defined ones. If the ray crosses an intersection between two regions in an approximately normal direction, it finds a match and then performs certain tasks of classification and reconstruction at the intersection. Depending on the application, the rays can be programmed to detect certain specific intersections and perform certain specific tasks. This leads to a very fast and effective segmentation approach which can successfully get rid of the partial volume effect.

The following is a brief description of the steps involved in our algorithm. We leave the implementational details to the next section where we show the application of our algorithm to solve the segmentation problems in virtual colonoscopy. The next section also describes how we automated each of these steps for the given application.

Approximate intensity based classification

In this step, we approximately classify the intensity values in the histogram of the dataset into distinct regions. The classification depends on the number and type of distict regions present in the dataset. An important point to note is that the region boundaries in the histogram are defined by approximate thresholds which are flexible. The unique intensity profiles at different intersections are then studied and stored.

Region growing

The algorithm then takes a seed point lying inside a region which the user wants to segment. From this seed point, we use the naive region growing to mark all the voxels that belong to the selected region and which are connected to the start voxel, untill we reach voxels that no longer belong to the selected region.

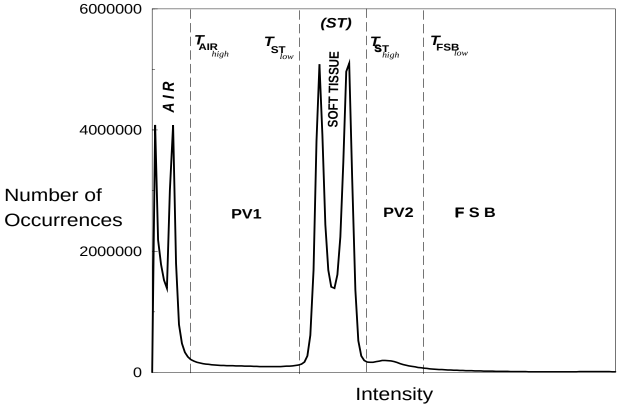

Figure 6: *Histogram of the CT dataset.*

Selecting starting points for segmentation rays

Before we go on to cast segmentation rays, we need to select voxels from which these rays emanate. The selection of these voxels is critical to the overall speed of the algorithm. The fewer the voxels, the faster the algorithm. Also, the nearer these voxels are to the intersection of different regions, the faster the rays detect intersections. The simplest and fastest strategy would be to assign the boundary voxels of the selected region as the starting points.

Detecting intersections using segmentation rays

After the starting points are selected, they are queued in. One by one these points are removed from the queue and from each point, segmentation rays are cast in all possible directions. The rays are always directed in one of the 26-connected-neighbor directions. We check which of the neighboring voxels does not belong to the selected region and cast a segmentation ray in that direction. Once a ray detects an intersection, it performs its given task and quits. The rays which do not find any intersection after traversing a certain distance are stopped and ignored.

Volumetric contrast enhancement

In the final step, the unwanted materials are removed from the volume by applying a programmed transfer function. This process is similar to contrast enhancement in image processing. Using a smooth transfer function, we get unaliased boundaries which improve the quality of volume rendering significantly.

4 Digital Colon Cleansing

In this section we explain an example application of our algorithm. We use our algorithm for digitally cleansing the colon for virtual colonoscopy. We now show how we implemented each step of our algorithm and how we automated each of those steps to get a fully automatic segmentation.

4.1 Approximate intensity-based classification

A closer look at the histogram obtained from the dataset immediately reveals two distinct-intensity regions (Figure 6). *Region 1*, which we call the *AIR* region, is on the lower end of the histogram. Voxels with intensities lying in this region are those that belong to air (from a prior knowledge of the CT intensities). This region can be characterized by just one threshold, $T_{AIR\,high}$. *Region 2* includes intensities that make up the center of the histogram. We name this region the *ST* (for Soft Tissue) region. This region is characterized by two thresholds, $T_{ST\,low}$ and $T_{ST\,high}$. The voxels with these intensities belong to the soft-tissue. Along with soft-tissue, this region also has some *PVE* voxels (some voxels at the intersection of the *AIR* and high intensity regions belong to this region). There is also a third region, *Region 3*, which makes up the

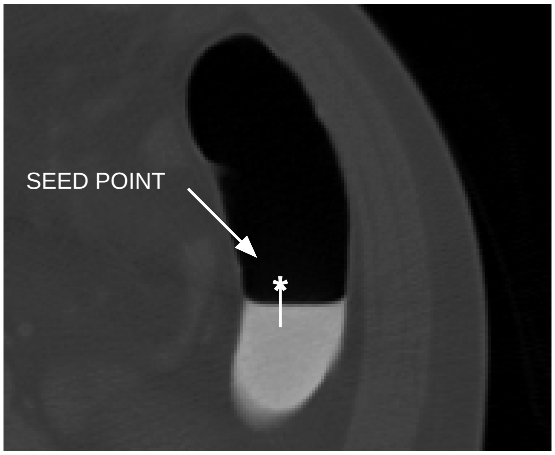

Figure 7: *Automatic location of seed points.*

higher end of the histogram, and includes voxels which belong to fluid, stool and bones. We name this region *FSB* (for Fluid, Stool and Bone) and characterize it by a single threshold level, $T_{FSB_{low}}$. Thus, we choose to broadly classify the volume into three main regions.

We name the two regions which lie in between the three regions defined in the above classification as *PV1* and *PV2* (for *P*artial *V*olume voxels). *PV1* is characterized by two thresholds, $T_{AIR_{high}}$ and $T_{ST_{low}}$. It includes the colon mucosa voxels (mucosa is the thin layer on the interior side of the colon) and some *PVE* voxels. Similarly, *PV2* is characterized by two thresholds $T_{ST_{high}}$ and $T_{FSB_{low}}$. This region includes *PVE* voxels along with some stool voxels whose intensities are lowered because of the neighboring lower intensity voxels.

The thresholds described above are automatically calculated from the histogram. The first two peaks in the histogram constitute the *AIR* voxels and hence we set $T_{AIR_{high}}$ to the right of the second peak. The next two peaks are the *ST* voxels and we set $T_{ST_{low}}$ and $T_{ST_{high}}$ around them. Due to the lack of a peak in the histogram for *FSB* voxel intensities, we use prior knowledge of the contrast enhancing fluid and its intensity and assign it to $T_{FSB_{low}}$. Since our algorithm does not depend on exact threshold values, the choice of above thresholds is flexible.

4.2 Region growing and start points for segmentation rays

To detect and mark the interior *AIR* region of the colon, we need seed points that are definitely inside the colon. We have devised a simple strategy to automatically and accurately locate seed points. The basic idea is to look for an *FSB* region voxel and check if there is *AIR* immediately above it. If this condition is satisfied, one of the *AIR* voxels is stored as a seed point. This strategy works accurately because the fluid, which is inside the colon, forms a smooth horizontal surface due to gravity (Figure 7). Hence, there is air on top of the fluid, and this air is definitely inside the colon. Any point inside this air can be the seed point for our algorithm. The only other high intensity material which is outside the colon, the bone, is surrounded by soft-tissues and the above condition will fail.

The above strategy is implemented by going through each vertical scan-line and checking for the above condition. If a seed point is found, it is stored in an array. In the end, we have all possible seed points in the array. Next, we use some of these seed points to mark the interior of the colon by region growing.

We now use region growing to get the boundary of the *AIR* region. We use the naive dilate-and strategy to do the region growing. To make this computationally efficient, we use a queue to store the voxels whose neighbors have not been explored. We start the region growing by placing the seed point in the queue. Now we explain the naive dilate-and strategy we use.

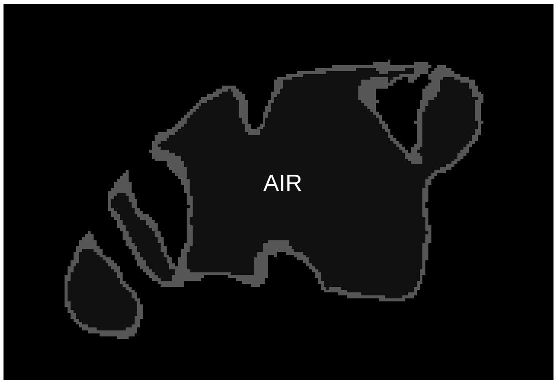

Figure 8: *AIR boundary detected by region growing.*

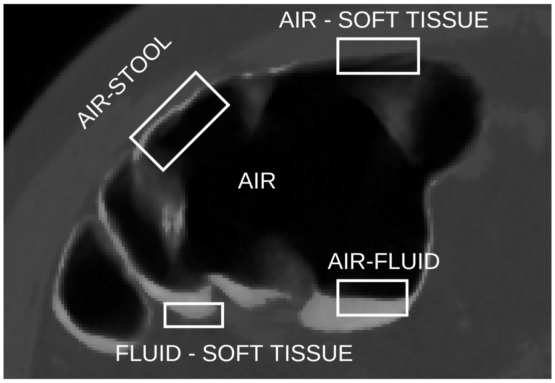

Figure 9: *Four different types of intersections.*

In each iteration, the first voxel in the queue is removed and its neighbors are inspected. If a non-masked *AIR* voxel is found, it is inserted into the queue; if a masked *AIR* voxel is found, it is ignored; and if a non-*AIR* voxel is found, then the current voxel is marked as a *boundary* voxel and is stored in a *boundary voxel (BV) queue*. After all the neighbors are inspected, the current voxel is removed from the queue and is masked to indicate that it is processed. The iterations end when there are no more voxels left in the queue. We use the voxels in the *BV queue* as the starting point for segmentation rays. This gives the best results because the intersection of *AIR* with other regions is in the immediate neighborhood of these voxels.

We use a 6-connection for detecting a neighboring, non-masked, *AIR* voxel, but a 26-connection for checking the boundary condition. A 6-connection gives smaller queue lengths and is thus more efficient. A 26-connection for boundary detection increases the accuracy of our algorithm (Section 4.3).

4.3 Detecting intersections using segmentation rays

Our next step exploits the peculiar characteristics at the intersection between *AIR* and the other regions, namely, *ST* and *FSB*. It is at these intersections that most of the *PVE* voxels occur and this step aims at removing them. This is an important step as it removes all the thin stool deposits and gives an improved contour to the colon, which is critical to the detection of the polyps.

From the previous step (Section 4.2) we get all the voxels lying on the rough contour of the colon (Figure 8) in the form of the *BV* queue. All the voxels which surround this *AIR* region belong to one of the following three types of intersections (Figure 9) :

1. *AIR* - *ST* intersection.
2. *AIR* - *Fluid* intersection.
3. *AIR* - *Stool* intersection.

We now differentiate between Fluid and Stool for two reasons. Firstly, since bone is not in the picture anymore, we cannot refer to the intersection as *AIR* - *FSB*. Secondly, a thin layer of stool is attached to the colonic mucosa at many places (Figure 9) and we choose to refer to such intersections as the *AIR* - *Stool* intersection. By *Fluid* we refer to relatively large high-intensity deposits which are mainly fluid (Figure 9).

Characterizing the intersections
We now need to characterize these intersections in order to distinguish between them. The characteristic properties were formed from a careful study of the intensity profiles at each intersection as we moved in a direction normal to the intersection. We show here the general intensity profiles at each of the intersections and note their characteristics.

The AIR - ST intersection

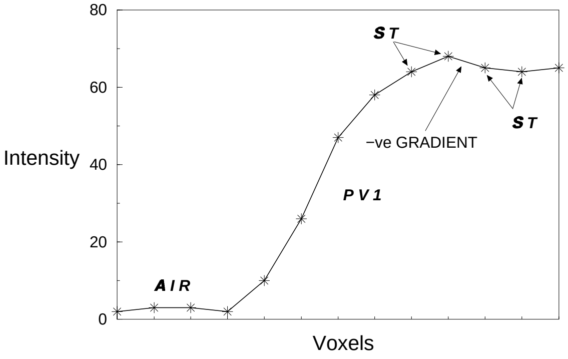

Figure 10: *Intensity profile at AIR - ST intersection.*

In Figure 10 we show the intensity profile at the *AIR* - *ST* intersection. From a careful analysis of this intensity profile we characterized the *AIR* - *ST* intersection by the following properties :

- A gradual increase in the gradient as the intensities move from *AIR* to *PV1* to *ST* and possibly to *PV2*.
- After the first zero or negative gradient, the intensity value is still in the *ST* region.
- The intensities never reach the *FSB* region.
- No more than one *PV2* region voxel is present before the negative or zero gradient.

The AIR - Fluid intersection
Figure 11 gives the intensity profile at the *AIR* - *Fluid* intersection. The following characteristics were associated with this intersection :

- A sharp rise in the gradient as the intensity values go from *AIR* to *FSB*.
- An *FSB* voxel is reached within 3 voxels after the last *AIR* voxel.

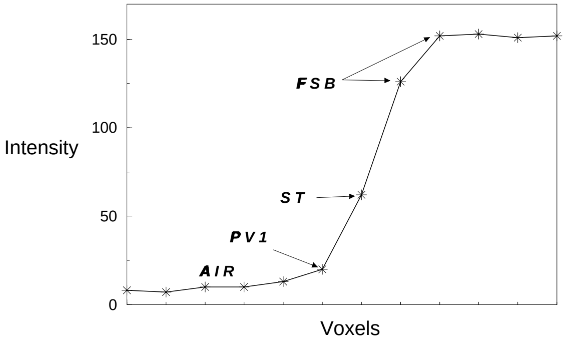

Figure 11: *Intensity profile at AIR - Fluid intersection.*

- The first negative or zero gradient does not bring the intensity value below the *FSB* range.

The AIR - Stool intersection

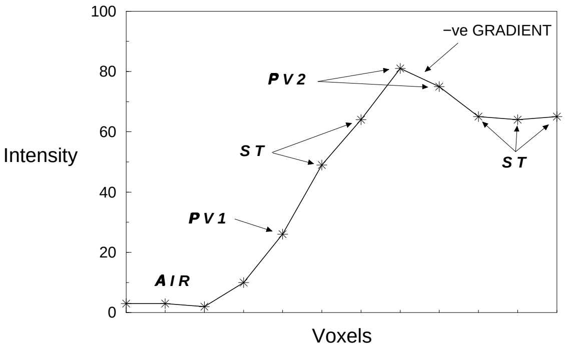

Figure 12: *Intensity profile at AIR - Stool intersection.*

From Figure 12 we characterize this intersection by the following properties :

- The intensity values sharply increase from *AIR* to *PV2* or *FSB*.
- After the first negative or zero gradient the intensity value lies in *PV2*.

An intersection is said to be found, when all the properties which characterize it are satisfied. Our next step uses these characteristics to detect intersections.

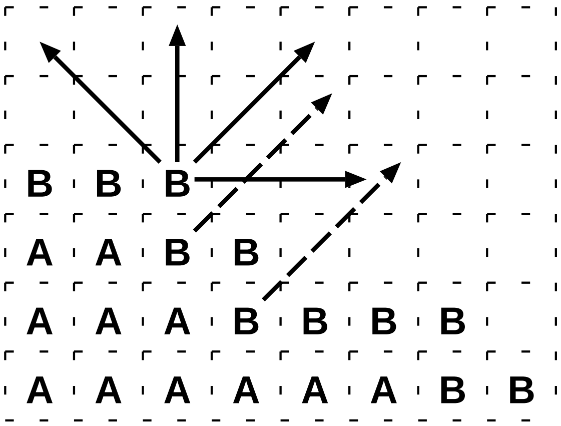

Figure 13: *Segmentation rays being sent to detect intersections. The dashed rays are not cast if we use 6-connection boundary checking. (A : AIR voxels, B : AIR boundary voxels.)*

Casting segmentation rays

We cast segmentation rays from each of the *AIR* boundary voxel (stored in the *BV* queue), outwards, in search of a matching characteristics. If the ray detects an intersection based on the above properties, then we perform a certain task that is associated with each type of intersection, or else the ray is simply ignored.

To determine the direction of each *segmentation ray* originating from a voxel, we examine all its 26-connected neighbors. The ones that are marked as *AIR* are ignored and we cast rays in the direction of all other neighbors. For simplicity, we show this in 2-D (Figure 13). The voxels marked *A* are the *AIR* voxels and those marked *B* are the *AIR* boundary voxels. We show rays sent from only one boundary voxel for clarity. Among all the rays, the ones that are most perpendicular to the intersection will most definitely detect it; others will be simply ignored.

In Section 4.2 we mentioned that we use 26-connection for checking if the *AIR* voxel is on the *AIR* boundary. By doing so, we avoid holes that would otherwise appear due to the lack of rays in some parts. To understand this, consider the dashed rays in Figure 13. The voxels from which they originate will not be marked as the boundary voxels if we use 6-connection boundary checking, thus leaving holes.

We now discuss the tasks we perform when a ray detects an intersection. The tasks mainly involve classifying the voxels into one of the three main regions (Section 4.1) and altering the intensity values to what they would have been in the absence of the residual fluid and stool.

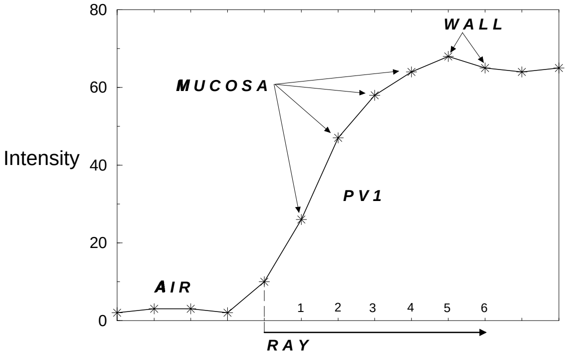

Figure 14: *Classification of voxels on the AIR - ST intersection.*

The AIR - ST intersection

In this case we mark the last two voxels of the ray as the *colon wall* voxels and mark the voxels preceding them as the mucosa voxels (Figure 14). Colonic mucosa lies on the interior of the colon wall and hence we use such a classification. An important point to note here is that we are not aiming at an accurate distinction between the colon wall and mucosa. We are only interested in an accurate inner contour of the colon which shows the polyps.

The AIR - Fluid intersection

Since our goal is removal of all *Fluid* voxels, we remove all the voxels on this intersection, thus eliminating all partial volume voxels that resulted from the intersection. We remove the voxels by marking them as *AIR* voxels and assigning them an intensity value of $T_{AIR_{high}}$ (Figure 15). The removal of the remaining fluid voxels is fairly simple because their intensities are in the *FSB* region and will be dealt with in the last step of our algorithm.

The AIR - Stool intersection

Due to the presence of high-intensity stool voxels, the thickness of the colonic mucosa increases. To overcome this problem and reconstruct the mucosa, we carefully remove the stool voxels along the ray and move the mucosa voxels ahead to their correct position. We show one example of the reconstruction in Figure 16.

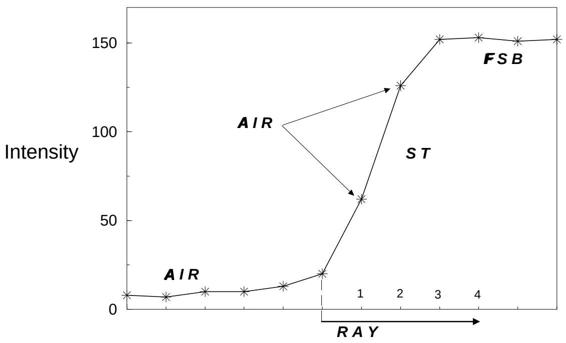

Figure 15: *Classification of voxels on the AIR - Fluid intersection.*

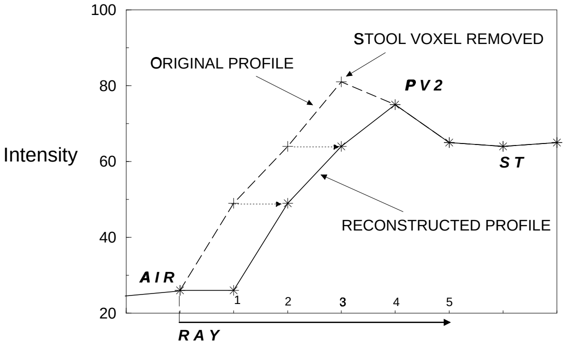

Figure 16: *Reconstruction and classification at the AIR - Stool intersection.*

By the end of this step of our algorithm, we have successfully removed all partial volume voxels that were in the *PV1* region and some that were in the *PV2* region. We are now left with all the high-intensity voxels of *FSB* range and the partial volume voxels on the *Fluid - ST* intersection. The final step handles these voxels.

4.4 Using volumetric contrast enhancement to reconstruct mucosa and remove fluid

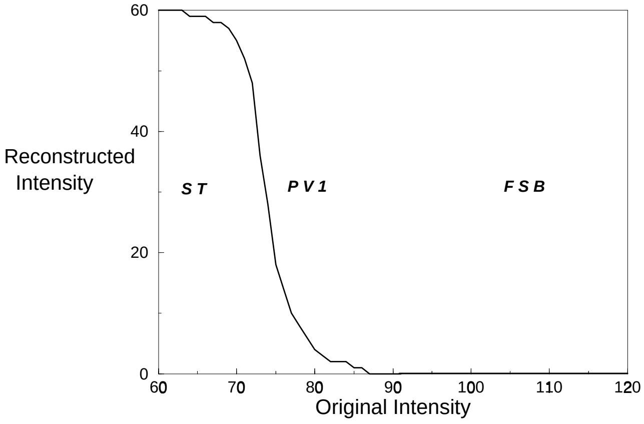

Figure 17: *Reconstruction at the Fluid - ST intersection.*

In the final step of our algorithm, we reconstruct the mucosa at the *Fluid - ST* intersection and remove the hight-intensity *FSB* voxels that are still remaining in the dataset. We use a technique similar to the *Contrast Enhancement* used in image processing[1]. Since we extend it to a volume, we call it the *Volumetric Contrast Enhancement*. We show the contrast enhancing reconstruction graph in Figure 17.

The key idea behind this graph is that if a voxel on the *Fluid - ST* intersection has an intensity closer to the *FSB* region, then the probability of that voxel being an *FSB* voxel is very high. The lower intensity of the probable *FSB* voxel is attributed to the effect of the surrounding low intensity voxels. Thus, the closer the voxel intensity is to the *FSB* region, the lower the reconstructed intensity will be (since we are aiming to remove the FSB voxels). Similarly, a voxel with an intensity closer to the *ST* region will have a reconstructed intensity closer to the average *ST* intensity. This reconstruction graph can be implemented by a simple lookup table taking just one pass through the dataset. Thus, this step is extremely fast. We prefer the curve over a straight line to increase the contrast between the two regions without aliasing effects. This results in smoother and more accurate images from volume rendering.

Figure 18: *Reconstructed mucosa at a Fluid - ST intersection after segmentation.*

5 Results

Our segmentation algorithm was tested for the virtual colonoscopy system and now, is an integral part of its preprocessing pipeline. By doing automatic histogram classification and seed point detection, the algorithm gave a fully automatic solution to segmentation and digital colon cleansing. We tested the algorithm on a variety of datasets. We use one of the datasets of size 512x512x411 to demonstrate the results. Since the patient did not undergo any physical colon-cleansing prior to the scan, there was a significant amount of residual fluid inside the colon. A thin layer of stool was found to be attached to the colonic mucosa in many places.

Our algorithm removed all the fluid and stool voxels accurately (Figures 19 and 20). Our method was able to detect and remove stool deposits as thin as one voxel thick which is about 0.7 mm. We were also able to unearth all the mucosa voxels lying below the fluid at the intersection of fluid with the colon wall. Due to the mucosa voxels, there was no aliasing effect at the newly detected interior surface of the colon (Figure 18).

Since the crux of our algorithm - characterizing the intersections - is done by our cognitive skills, we claim that our algorithm gives just as accurate a result as a manual segmentation. In fact, since we do not miss even a single intersection, our algorithm could be better than manual segmentation.

The total time for the whole segmentation process for the selected dataset was 58 seconds on an SGI Onyx2 (R10000, 195MHz) and 35 seconds on a Linux PC (AMD Athelon, 750 MHz). This is the fastest time reported for colon segmentation which also includes digital cleansing.

This paper showed the application of our segmentation algorithm for virtual colonoscopy. The algorithm, or its parts, could be used for any application requiring a similar complex segmentation. It will be especially useful when there are partial volume effect concerns.

6 Conclusion and Future Work

In this paper we have described an algorithm for fast and accurate segmentation with the ability to remove the partial volume effect. This algorithm is a very general algorithm which can be used by any application with segmentation requirements similar to those of virtual colonoscopy.

In the future work, we plan to build an interactive segmentation system based on our algorithm in which the user can interactively configure all the parameters required by the algorithm. This will include the ability to pick intersection characteristics using a mouse and assigning classification/reconstruction tasks to the rays that find a perticular intersection. Since the algorithm is extremely fast, we also plan to add visual feedback by rendering and displaying the segmented dataset after each operation by the user.

Acknowledgements

This work has been supported by grants from NIH #CA79180, ONR #N000149710402, and Viatronix Inc. The patient's datasets were provided by the University Hospital of the State University of New York at Stony Brook. Special thanks to Klaus Mueller and Kevin Kreeger for the encouraging discussions. We thank Frank Dachille and Kathleen McConnell for their helpful comments on the draft of this paper.

References

[1] R. Gonzales and R. Woods, Digital Image Processing, Addison-Wesley 1992.

[2] K.H. Hohne and W. Hanson. Interactive 3D Segmentation of MRI and CT Volumes using Morphological Operations, *Journal of Computer Assisted Tomography*, **16**(2):285–294, March 1992.

[3] L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang, 3D Virtual Colonoscopy *Proc. Symposium on Biomedical Visualization*, pp. 26–32, 1995.

[4] L. Hong, S. Muraki, A. Kaufman, D. Bartz, T. He, Virtual Voyage: Interactive Navigation in the Human Colon, *Proc. SIGGRAPH '97*, pp. 27–34, 1997.

[5] Z. Liang, F. Yang, M. Wax, J. Li, J. You, A. Kaufman, L. Hong, H. Li, and A. Viswambharan, Inclusion of *A Priori* Information in Segmentation of Colon Lumen for Virtual Colonoscopy, *Conf Record IEEE NSSS-MIC, CD-ROM*, 1997.

[6] Z. Liang, D. Chen, R. Chiou, B. Li, A. Kaufman, and M. Wax, On Segmentation of Colon Lumen for Virtual Colonoscopy, *Proc. SPIE Medical Imaging*, 1999.

[7] W. Lorensen, F. Jolesz, and R. Kikinis, The Exploration of Cross-Sectional Data with a Virtual Endoscope, *Interactive Technology and the New Medical Paradigm for Health Care, eds. R. Satava and K. Morgan*, pp. 221–230, 1995.

[8] J. Mandel, J. Bond, J. Church, and D. Snover, Reducing Mortality from Colon Cancer Control Study, *New England J Med 328*, pp. 1365–1371, 1993.

[9] M. Wan, Q. Tang, A. Kaufman, Z. Liang, and M. Wax, Volume Rendering Based Interactive Navigation Within The Human Colon, *Proc. IEEE Visualization '99*, pp. 397–400, 1999.

[10] M. Wax, Z. Liang, D. Chen, B. Li, R. Chiou, A. Kaufman, and A. Viswambharan, Electronic Colon Cleansing for Virtual Colonoscopy, 1^{st} *Intl Conference on Virtual Colonoscopy, Boston, MA*, October 1998.

[11] C.L. Wyatt, Y. Ge, and D.J. Vining, Automatic Segmentation of the Colon, *Proc. SPIE Medical Imaging*, 1999.

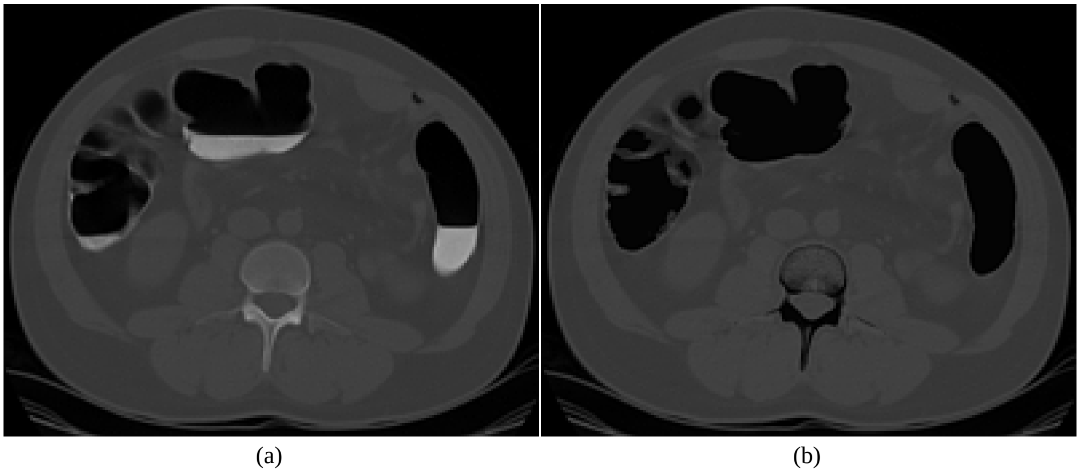

Figure 19: A cross-section of the CT data showing colon (a) before segmentation, and (b) after segmentation.

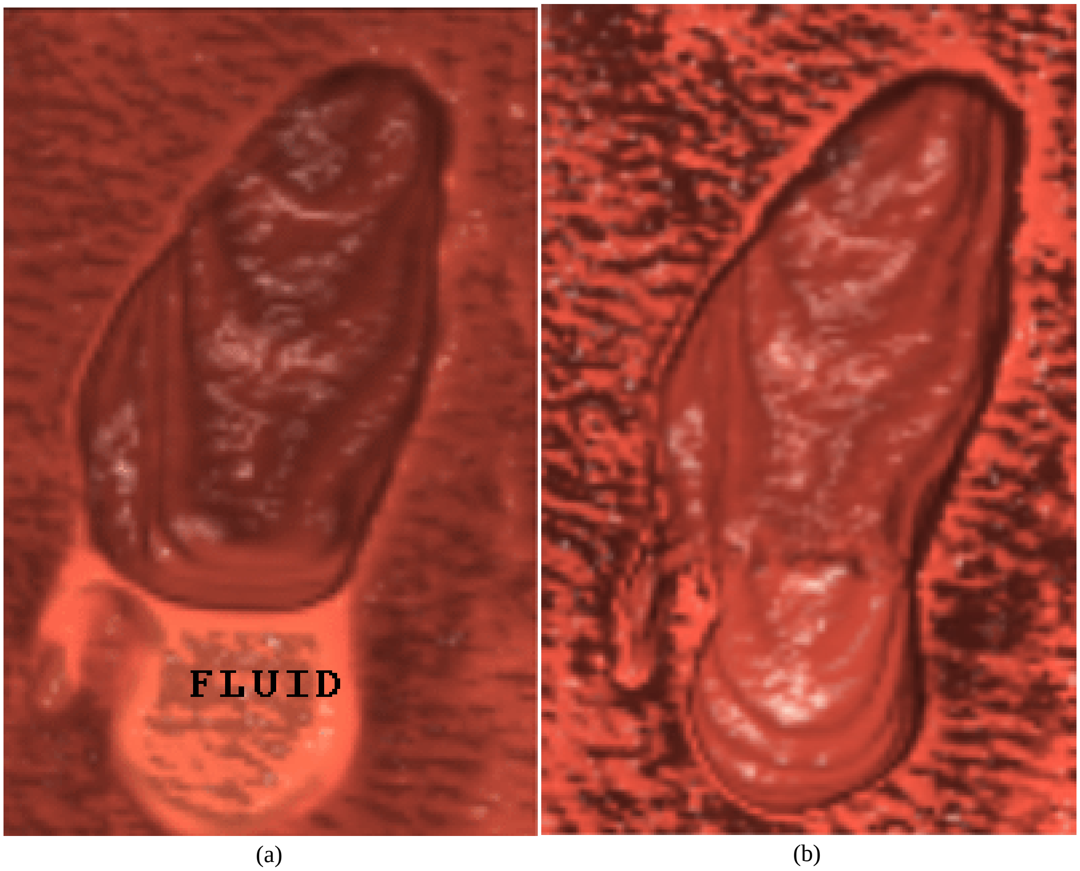

Figure 20: Volume rendered images showing (a) presence of fluid before segmentation, and (b) fluid removed after segmentation.