

FURTHER ACCELERATION OF THE VISUALISATION PROCESS USED IN VIRTUAL COLONOSCOPY

Mehran Sharghi and Ian W Ricketts
Department of Applied Computing
University of Dundee
Dundee, DD1 4HN, Scotland, UK.
[msharghi/ricketts]@computing.dundee.ac.uk

ABSTRACT

We present a novel technique to accelerate the volume ray casting process used in virtual colonoscopy. Virtual colonoscopy is a less invasive alternative to colonoscopy with potential for wide use in the early detection of colorectal cancer. The idea of our method is to find the exact distance from each image plane pixel to the closest colon wall (boundary) voxel and use this distance information to start ray integration directly from the colon boundary. Our method also improves the rendering time by exploiting the fact that only a small part of the colon is visible from any internal viewing point. We apply a pre-processing visibility determination algorithm to identify the potentially visible part of the colon at each internal view point. The algorithm only projects the potentially visible boundary voxels on to a distance buffer. Using this distance it is possible to skip the space and start the ray casting integration directly from the colon wall. We have used parallel projection and a generic projection template to achieve more acceleration in our implementation. The method has been implemented on an IBM compatible personal computer and tested with a synthetic colon data set.

KEY WORDS

Virtual colonoscopy, visualisation, volume ray casting, acceleration, space leaping, and visibility.

1. INTRODUCTION

Colorectal cancer is reported as the second most common cause of death, after lung cancer, in the US, Canada [1], and the UK [2]. According to the American Cancer Society, over 130000 new cases of colorectal cancer are diagnosed each year in the US and over 56000 die as a result [1]. Early detection of colorectal cancer significantly increases the chances of survival. Unfortunately, early-stage colorectal cancer usually produces few if any symptoms and therefore the current strategy for early detection relies on population screening. It has been reported that colon cancer mortality can be decreased to one third by regular screening [3]. Annual

testing for colorectal cancer is recommended for everyone over the age of 50 according to the American Cancer Society guidelines and the National Cancer Institute recommends regular screening for all people over the age 40. Unfortunately, currently available screening methods are either weakly specific and sensitive, or invasive and expensive. Colonoscopy is currently the most sensitive and specific screening method, however it is invasive and not completely reliable [4, 5].

Virtual colonoscopy was introduced by Vining et al [6] as a less invasive and cost effective alternative to colonoscopy. In this method the intra-lamina view of the colon as it is visible through a colonoscope is visualised by reconstructing 3D images from cross-sectional images (e.g. CT scan images) of the abdomen. There is a possibility to virtually navigate within the colon and search for abnormalities. This procedure usually involves four stages: patient preparation, image acquisition, visualisation, and interpretation of the results.

Successful use of virtual colonoscopy relies on high quality three-dimensional visualisation. It has been reported that virtual colonoscopy using volume rendering is more sensitive and specific compared to the use of surface rendering [7]. Furthermore, when a suspected abnormality is found, volume rendering enables the clinicians to closely examine both the tumour and its underlying tissue. However, the significant requirement for computational resources in volume rendering prohibits the interactive use of it. A variety of methods have been proposed to accelerate volume rendering including hierarchical structures (pyramid), space leaping, image and object space coherency, frame coherency, etc. These are reviewed by Yagel et al [8, 9] and Danskin et al [10].

Space leaping which involves the efficient traversal or skipping of empty space is a general technique to accelerate volume ray casting. The advantage of this technique is that it does not degrade the quality of the visualisation and is well matched to applications such as virtual colonoscopy where the image plane is within a space surrounded by the object. Any method which traverses the empty space in an efficient way could be classified as a space leaping technique. There have been

reported methods to accelerate ray casting as it is used in virtual colonoscopy by applying space leaping. Some of these methods are reviewed below. You et al [11] applied the polygon assisted ray casting technique (initially introduced by Avila et al [12]) to accelerate volume rendering in virtual colonoscopy. They used a surface rendering algorithm to locate the inner bounding box of the colon. Then during the ray casting the intersection point of each ray and a polygon representing the colon wall is calculated to start the ray integration directly from the colon wall. Wan et al [13] applied the distance transform to accelerate this in virtual colonoscopy. This idea was first used by Zuiderveld et al [14] to accelerate the ray casting process. They calculated the distance transform for all space voxels within the colon. During ray casting, from any starting voxel within the colon cavity, rays can be extended in a single step by the value of the distance transform of that starting voxel. Since the distance transform value for any voxel is the minimum distance to reach the colon wall it is guaranteed that the ray will not intersect with any colon wall voxel. Once the ray reaches the colon wall the normal ray casting process is used. Vilanova et al [15] used cylinders to approximate the space within tubular shaped organs (e.g. colon, blood vessels, etc). Then in the ray casting phase the intersection of rays and these cylinders are calculated to traverse the space in larger steps. The cylinders are fitted along the centre line of the organ and wholly within it. The diameter of the cylinders is determined using the distance transform values of the voxels along the centre line. Sharghi et al [16] approximated the space inside the colon by spheres and then in the ray casting process used this approximation to traverse the space more efficiently. Their approximation is based on the skeleton of the colon cavity.

2. OVERVIEW OF THE METHOD

This paper reports on the use of space leaping to accelerate visualisation via volume ray casting. It is particularly well suited to virtual endoscopy in which the image plane is within a hollow organ. Space leaping seeks to rapidly traverse any space without adversely affecting the quality of the rendering. Our method endeavours to integrate along each ray starting at the object boundary leaping over space voxels. The general idea of the method is to find the exact distance from any image plane pixel to the closest non-space voxel (i.e. boundary voxel) which intersects with the ray cast from that image plane pixel, then to use this distance to start ray casting from an object boundary voxel. One approach is to find this distance by projecting all boundary cells on to the image plane. The actual projection is performed on to a z-buffer where only the depth information of the projected cells is kept. In this paper we refer to this z-buffer as the distance buffer. The overhead of projecting all boundary cells to produce each rendered image makes the method less attractive as an acceleration method. Wan et al [17] used a projection

template to reduce the amount of computation involved in projecting cells on to the distance buffer. Our method further improves the technique by exploiting a visibility algorithm devised for virtual colonoscopy. It relies on the assumption that for virtual endoscopy applications in general and virtual colonoscopy in particular, only a small part of the organ is usually visible from any internal viewpoint. It is therefore not necessary to project all boundary cells. Our method uses a pre-processed visibility determination technique and considers only a potentially visible subset of boundary cells. Details of this visibility determination technique are presented in the next section.

3. VISIBILITY DETERMINATION

The visibility information required in virtual endoscopy determines which boundary voxels are visible from the image plane. There are two general approaches to determining visibility. One approach is to determine visibility interactively, i.e. calculating the visible set before rendering each frame. The advantage of this method is that it is possible to produce the exact visible set for the known viewpoint. The disadvantage of the method is the overhead of computations required to render each frame. Another approach is to determine visibility in a pre-processing stage. The advantage of this latter approach is the elimination of the interactive visibility determination computations, and the cost is sending a more conservative (i.e. larger) potential visible set to the next stage of the rendering pipeline.

Dealing with single voxels to determine the visibility could be computationally inefficient. One solution is to divide the object in to sections and determine the visibility between sections. This is particularly relevant when the visibility is determined in a pre-processing stage. Consider the architecture of a house. Assume that each section is taken to be a room in the house. This subdivision into sections results in a conservative visible set because a section/room may only be partially visible (i.e. contains voxels which are not visible) but we have to consider it as a wholly visible section/room. In the colon there are no obvious section boundaries. The usual approach is to divide the colon into consecutive sections. When the image plane is inside one section, only those sections visible from that section need to be considered. It is also important to divide the object into an optimal number of sections. On the one hand a large number of shorter sections requires excessive computation to determine visibility between sections. On the other hand a small number of larger sections produces an overlarge potential visible set, i.e. much larger than is strictly required.

Our visibility determination algorithm is specifically designed to take advantage of the colon's anatomy. The algorithm consists of two steps. The first step is to divide

the colon into sections and the second step is to determine the visibility between sections.

We need to clarify the concept of a cell which we use in this paper. When a ray intersects with the object boundary the intersection point does not normally coincide with the exact centre of a voxel. Consequently, at the intersection point eight surrounding voxels must be considered. These voxels are the vertices of a cube representing the closest integer coordinate to the intersection point in 3D space. We call this cube of eight voxels a cell. If all voxels of a cell are space voxels then we refer to the cell as a space cell and if any of these voxels is not space then we call the cell a non-space cell.

3.1. Dividing the colon into sections

Our algorithm divides the colon into sections based on the curvature of the colon's centre line. The colon's centre line is a single voxel path in 3D space from one end to the other of the colon. There are no branches or bifurcations in the centre line. The algorithm starts by finding the centre line of the colon. It then smoothes the centre line to prevent sharp changes along it. The algorithm then starts from one end of the centre line and accumulates the curvature of it. When the accumulated curvature reaches a preset threshold the algorithm searches for the best place to end the section. The end of a section is specified by a cut plane whose boundary is the colon wall and is preferably perpendicular to the colon's local direction. We call this cut plane a portal. The algorithm performs the search to find the end of section in a portion of centre line where the accumulated curvature is between a lower and an upper threshold. Portals of minimum area, i.e. where the colon is narrowest are the best place to end a section. However, finding these portals is not computationally efficient. We used an approximate method based on the distance transform value at the colon centre line voxels. The algorithm searches in the portion of the centre line limited by the accumulated curvature thresholds. The voxel with the minimum distance transform value is chosen as the end of section. The next step is to specify a portal which passes through this voxel and is perpendicular to the local direction of the centre line. Voxels belonging to this portal are determined using a 2D region growing algorithm in the portal plane. These voxels are stored in a special data structure for later use. Figure 1 illustrates a synthetic colon data set divided into sections using the above algorithm.

The algorithm also identifies the voxels inside each section including one of its ending portals by using a 3D region growing algorithm. A section identity is stored in each of these voxels for later use during the projection and ray casting phases.



Figure 1 – The colon divided into sections

3.2. Determining visibility between sections

After dividing the colon into sections the next step is to determine the visibility between sections, i.e. to determine which sections are visible from any particular starting section. It is not necessary to examine the visibility between every possible pair of voxels in the two sections in order to determine the visibility between sections. If a voxel in one section is visible from a voxel in another section then we consider both sections to be visible from each other.

Our algorithm uses the following method to solve the problem in an efficient way. Consider that we are trying to determine those sections which are visible from section A in figure 2. Section B, as a neighbour, is always visible from section A. To examine the visibility between sections A and C our algorithm examines the visibility between the portals ending each section. In our algorithm section C is considered visible from section A if portal P2 is visible from portal P1. If section C is not visible from section A, the algorithm terminates. If section C is visible

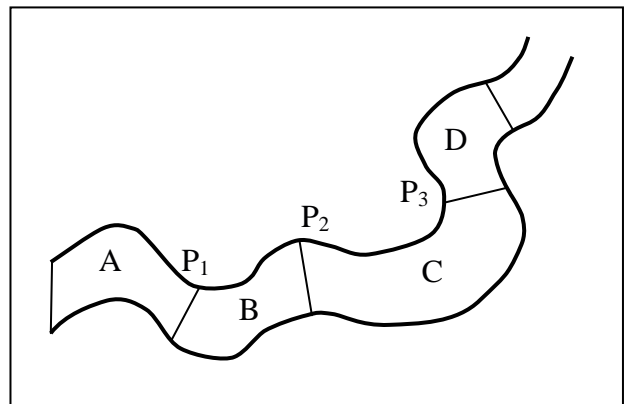


Figure 2 – Determining visibility between sections

the algorithm continues further and examines the visibility between portal P1 and P3. In order to determine visibility between portals the algorithm checks the visibility between their voxels. If any of the voxels of a portal is visible from one of the voxels of another portal then these two portals are considered visible from each other. The visibility between two voxels v_1 and v_2 is determined by casting a ray from v_1 toward v_2 . If the ray intersects any non space cell before reaching v_2 then the two voxels are considered as not being visible to each other. This is computationally efficient because the number of voxels in a portal is much less than the number of voxels in a section.

The visibility information is stored in an array with two entries for each section. The corresponding entry for each section in the array specifies how many sections are visible before and after that section.

4. PROJECTION AND RENDERING

Remember that the image plane will be inside the volume data set so we must identify the section(s) in which the image plane resides. Our method first identifies the voxels corresponding to the four corners of the image plane. Then using the section identity stored in these voxels it identifies the section(s) where the image plane is located.

Using the pre-processed visibility information we are able to identify the visible section from the current position of the image plane. The next step is to project the boundary voxels of the visible sections and section(s) where the image plane is located, onto the distance buffer. The boundary cells of each section are identified and stored in a data structure in the pre-processing step. We use a list of sections and store the boundary cells of each section in an array as illustrated in figure 3. This facilitates efficient access to the boundary voxels of each section during the projection phase.

A distance buffer (z-buffer) is used in the implementation of our method and it contains one entry for each image plane pixel. Values in the buffer indicate the distance from the corresponding image plane pixels to the nearest non-space cell along a ray in the viewing direction. The distance buffer is initialised to a value that indicates the distance is not yet calculated. The distance buffer is updated by the projection of boundary cells from all visible sections before rendering each frame. The depth information for projected cells is stored in the distance buffer.

The projection of a cell, which is a cube in the volume, usually produces a hexagonal footprint. To project a cell it is enough to project six vertices (out of eight) onto the distance buffer plane and specify the entries in the distance buffer covered by the projection footprint. In our implementation we have only considered parallel

projection; however it is possible to extend the method to perspective projection. One important property of the parallel projection is that the projection of all cells regardless of their position in the volume will be of the same form and size. This property is used to increase the performance of the method. We used the projection template as described by Wan et al [17]. The projection process is performed once for a cell and a template projection footprint is constructed. This projection template specifies the pixels which are inside the hexagonal footprint. For each boundary cell only one point needs to be projected onto the distance buffer and the other points covered by the projection footprint are determined using the projection template. The projection template needs to be reconstructed whenever the viewing point or direction is changed.

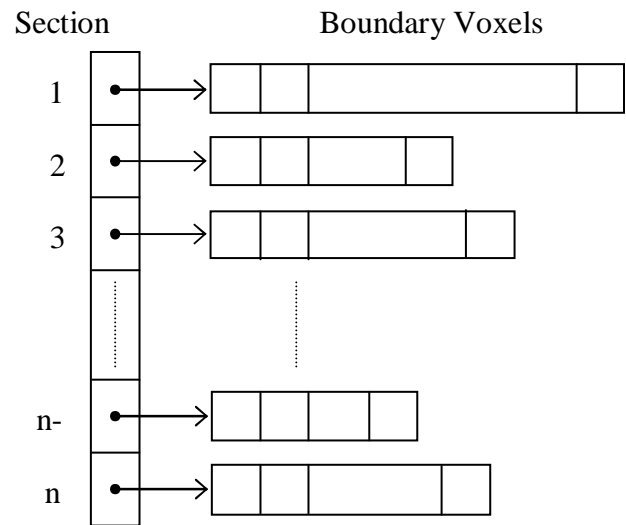


Figure 3 – Data structure to store boundary voxels

During the ray casting process, as the image plane is swept pixel by pixel, the corresponding entry in the distance buffer is examined. If the distance has already been calculated then its value is used to skip the space and start the ray casting from a boundary cell. If the distance has not been previously calculated it means that the ray does not collide with any boundary cell and therefore no ray casting is required.

5. RESULTS

We implemented the method on an IBM compatible personal computer. Testing used a computer generated synthetic colon data set of 200 slices with 250x150 pixels per slice. This data set is freely available from the authors. Among the pre-processing operations required in our method the most time consuming ones are the distance transform calculations for those voxels belonging to the colon and the space inside it, the colon centre line

extraction, and determination of the visibility between sections. Using the static data set we measured 4, 8, and 25 seconds approximate computation time for each of these operations respectively. The total pre-processing time was approximately 40 seconds. The interactive rendering time includes the boundary cell projection time and ray casting. The projection time is directly related to the number of boundary cells in the visible sections. There were, on average, three visible sections when we applied the method to the above data set. Furthermore, the average number of visible voxels at each section was 29800 which is approximately 28% of the total number of boundary voxels. We performed an animation based on the centre line of the colon and compared the rendering time of our accelerated method and a basic ray casting. Our method was 5 times faster.

6. CONCLUSION

We presented a method to accelerate volume ray casting via space leaping. The method is used in the visualisation phase of virtual colonoscopy. The general idea of our method is to find the exact distance from each image plane pixel to the closest non-space voxel (i.e. a boundary voxel) and use this distance to start integration along each ray directly from the object boundary. This distance information is obtained by projecting boundary voxels in to a distance buffer. We also applied a visibility determination method to reduce the number of projected voxels in order to further speed up the process. This method represents a significant improvement over that reported by Wan et al [17] through the use of a visibility determination algorithm. Furthermore, our visibility determination method also represents an improvement compared to the method used by Hong et al [18]. They used an interactive determination of visibility to reduce the number of triangles for the next stage in their surface rendering pipeline. Our approach is different to their method in that we used a pre-processing visibility determination approach and hence suffered no interactive overhead computation. Moreover, our method divides the colon into sections with portals perpendicular to the colon's centre line instead of restricting portals to be perpendicular to one of the principle axes.

We applied this method in a virtual colonoscopy, however this method is not limited to virtual colonoscopy or even virtual endoscopy. The projection of boundary cells can be used in any volume ray casting in which space leaping is applicable. The visibility determination technique is specifically designed for the colon and unless the new object has a similar form to the colon (i.e. a tubular, hollow object) it will need some modification in order to be applied more generally.

4. ACKNOWLEDGEMENT

The authors wish to acknowledge the financial support of the Ministry of Science, Technology, and Higher Education of Iran.

REFERENCES

- [1] Colorectal Cancer Facts and Figures, Colon Cancer Alliance, <http://www.ccalliance.org>, Last visited April. 2002.
- [2] Z. Kmiotowicz, NHS aims to improve care for colorectal cancer, *BMJ*, 315, 1997, 1485-1486.
- [3] D. Vergano, Annual Colorectal Cancer Screening Cuts Death Risk, *Medical Tribune*, 40(7), 1999, 19.
- [4] P. Pescatore, T. Clucker, J. Delarive, R. Meuli, D. Pantoflickova, B. Duvoisin, P. Schnyder, A. L. Blim, and G. Dorta, Diagnostic Accuracy and Interobserver Agreement of CT Colonography (Virtual Colonoscopy), *Gut*, 47, 2000, 126-130.
- [5] D.K. Rex, C.S. Cutler, G.T. Lemmel, E.Y. Rahmani, D.W. Clark, D. J. Helper, G.A. Lehman, and D.G. Mark, Colonoscopic Miss Rates of Adenomas Determined by Back-to-Back Colonoscopies, *Gastroenterology*, 112, 1997, 24-28.
- [6] D.J. Vining, D.W. Gelfand, R. Bechold, E. Sharling, E. Grishaw, and R. Shifrin, Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality, *Proc. Annual Meeting of the American Rontgen Ray Society*, 1994, 14.
- [7] K.D. Hopper, A.T. Iyriboz, S.W. Wise, J.D. Neuman, D.T. Mauger, and C.J. Kasales, Mucosal Details at CT Virtual Reality: Surface versus Volume Rendering, *Radiology*, 214, 2000, 517-522.
- [8] R. Yagel and Z. Shi, Accelerating Volume Animation by Space-Leaping, *Proc. of IEEE Visualization 93*, 1993, 62-69.
- [9] R. Yagel, Towards Real Time Volume Rendering, *Proc. of GRAPHICON'96*, Vol. 1, July 1996, 230-241.
- [10] J. Danskin. and P. Hanrahan, Fast algorithms for volume ray tracing, *1992 Workshop on Volume Visualization*, 1992, 91-98.
- [11] S. You, L. Hong, K. Junyaprasert, A. Kaufman, S. Muraki, and Y. Zhou, Interactive Volume Rendering for Virtual Colonoscopy, *Proc. of Visualization '97*, 1997, 433-436.
- [12] R. Avila, L. M. Sobierajaski, and A. Kaufman, Towards a Comprehensive Volume Visualization System. *Proc. of Visualization '92*, 1992, 13-20.
- [13] M. Wan, Q. Tang, A. Kaufman, Z. Liang, and M. Wax, Volume Rendering Based Interactive Navigation Within the Human Colon, *Proc. of IEEE Conference Visualization 99*, 1999, 397-400.
- [14] K. Zuiderveld, A. H. J. Koning, and M. A. Viergever, Acceleration of ray Casting Using 3D Distance Transform, *Proc. of Visualization in Biomedical Computing 92*, 1992, 324-335.

- [15] A. Vilanova, E. Gröller, and A. König, Cylindrical Approximation of Tubular Organs for Virtual Endoscopy, <http://www.citeseer.nj.nec.com/320682.html>, Last visited March 2001.
- [16] M. Sharghi and I.W. Ricketts, A Novel Method for Accelerating the Visualisation Process Used in Virtual Colonoscopy, *Proc. 5th International Conference on Information Visualisation*, London, July 2001, IEEE Computer Society, 167-172.
- [17] M. Wan, S. Bryson, and A.Kaufman, Boundary Cell-Based Acceleration for Volume Ray Casting, *Computers & Graphics*, 22(6), 1998, 715-721.
- [18] L. Hong, Z. Liang, A. Viswambharan, A. Kaufman, and M. Wax, Reconstruction and Visualization of 3D Models of Colonic Surface, *IEEE Transactions on Nuclear Science*, 18(3), 1997, 1297-1302.