Interactive visualisation of a virtual colonoscopy by combining ray casting with an acceleration corridor

Mehran Sharghi[a] and Ian W. Ricketts[b]

Applied Computing, University of Dundee, Dundee, DD1 4HN, Scotland, UK.

Abstract. We present a novel acceleration technique for the volume ray casting process used in virtual colonoscopy. Our method exploits the information obtained from casting a ray to avoid redundant computations for neighbouring rays. A corridor from the image plane into the volume data set is examined whenever a ray is cast. All rays passing through this corridor reach the same cell as the first non-space intersected cell. We apply the distance information obtained from casting a ray to other rays in the same group. Using this distance it is possible to skip the space and start the ray casting integration from a non-space cell. The group of rays is obtained by projecting the non-space cell intersected by the first ray in to the image plane. The method has been implemented on a personal computer and tested with the visible human data set and a synthetic colon data set. The amount of acceleration achievable using this method depends on the enlargement factor. There is some degradation in the quality of the image produced using this method, due to the approximation used to select the group of rays that reaches the same non-space cell.

1. Background

Colorectal cancer is the second most common cause of death, after lung cancer, in the US, Canada [1], and the UK [2]. The American Cancer Society estimated more than 148300 new cases of colorectal cancer will be diagnosed this year in the US and over 56600 will die as a result [1]. Colorectal cancer also causes approximately 17000 deaths in the UK each year [2]. Early detection significantly increases the chances of survival. According to one long-term study, regular screening for colorectal cancer can decrease mortality by one third [3]. Colonoscopy, though not completely reliable, is currently the most sensitive and specific method of colon examination [4, 5].

Virtual colonoscopy (CT colonography) was introduced by Vining et al [6] as a reliable, less invasive and cost effective alternative to colonoscopy. It uses cross-sectional (slice) images of the abdomen to produce an intra-lamina visualisation of the colon with the capability of ‘virtually’ navigating the colon to 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. Volume rendering via ray casting is the preferred method for producing these visualisations but requires substantial computing resources. Moreover, when a suspected abnormality is found, volume rendering enables the clinician to closely examine both the tumour and it’s underlying tissue and thereby assists in distinguishing between benign and malignant lesions [7]. The requirement for significant computing resources is a barrier to volume rendering being used in interactive visualisation systems where low rendering rates make it difficult for the clinician to control the viewing point. The aim of this work was to accelerate the rendering of virtual colonoscopy sufficiently to enable the potential users within surgery, radiology, and gastroenterology to interactively visualise and navigate 3D data on a standard personal computer.

A variety of methods have been proposed to accelerate the ray casting process and some of them are reviewed by Yagel et al [8, 9] and Danskin et al [10]. One group of acceleration techniques are termed space leaping, and involve the efficient traversal or skipping of empty space by the ray casting method. Space leaping offers the advantage that it does not degrade the quality of the visualisation and is well matched to the anatomy of the human colon which is a largely hollow organ. Some of the space leaping based acceleration techniques that have been applied to the task of virtual colonoscopy are summarised below. Sharghi et al [11] approximated the space inside the colon by spheres and then in the ray casting process this approximation was used to traverse the space efficiently. You et al [12] applied the polygon-assisted ray casting technique introduced by Avila et al [13], and used a surface rendering algorithm to locate the inside surface of the colon. The intersection point of each ray and a polygon representing the colon surface was then calculated to skip the space and to start from the colon surface directly. Zuiderveld et al [14] were the first to propose the use of the distance transform to accelerate ray casting and Wan et al [15] applied this to virtual colonoscopy. They used the distance transform, calculated for each voxel within the colon, to skip the space more rapidly.

[a] msharghi@computing.dundee.ac.uk

[b] ricketts@computing.dundee.ac.uk

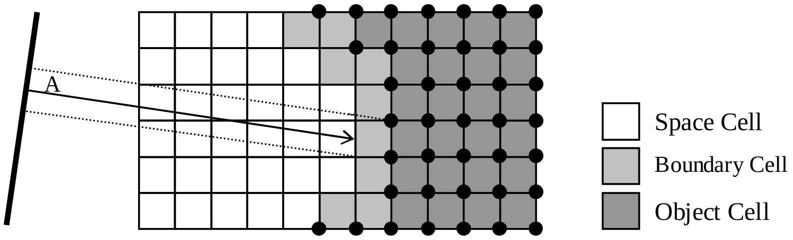

Figure 1. Corridor specified by casting ray A

2. Method

In the basic ray casting method the image plane is constructed pixel by pixel. For each pixel in the image plane a ray is cast to intersect with the volume along a line segment. Integration is performed on the voxels along this line segment to produce the intensity value of the corresponding pixel in the image plane. In this paper we present a novel method which can be classified as a space leaping technique to accelerate ray casting. As mentioned above the concept of space leaping is to rapidly traverse any space without adversely affecting the quality of the rendering. Our method exploits the information obtained from casting a ray to avoid redundant computations for a group of neighbouring rays. This information includes the distance between an image plane pixel and the closest non-space voxel which intersect with the ray cast from that image plane pixel.

Integration in the ray casting is performed at discrete points along each ray. These points do not normally coincide with the exact centre of a voxel. Consequently, at each point in 3D space where we examine the corresponding value, eight voxels that surround this point must be considered. These voxels are the vertices of a cube representing the closest integer coordinate to the 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 a space cell and if any of these voxels is not space then we call the cell a non-space cell. We choose one of these voxels as the cell reference and store a code in this voxel to indicate its space or non-space status. This is done for the whole volume in a pre-processing step. The code is used during the later rendering process to efficiently determine the space or non-space status of a cell.

Cells along each ray are examined until a non-space cell is reached. Examining a cell is equivalent to examining a cube-form subspace of the volume. Considering the whole examined path for a ray, it can be seen that a corridor from the image plane into the volume data set is tested for each ray. All rays passing through the same corridor reach the same non-space cell at the same distance from the image plane. Therefore, if one knows this distance, it is possible to skip the space and start the ray casting integration from the non-space cells.

In our method the distance to the first non-space cell is obtained by casting one of the rays from the group. This distance is then used for all the other rays of the group to skip the space and start the ray casting at the non-space cell. It is computationally inefficient to determine the exact group of rays which pass through this corridor, so we have used an approximation. In our method the group of rays is obtained by projecting the non-space cell, intersected by the first ray, back on to the image plane. Those rays corresponding to the pixels covered by the projection footprint form the group of rays. Figure 1 illustrates a corridor specified by casting ray A and projecting the intersected boundary cell on to the image plane in two dimensions.

2.1. Projecting cells on to the image plane

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 the same form and size. This property is used to increase the performance of the method. The projection of a cell, which is a cube in the volume, usually produces a hexagon on the image plane. To project a cell it is enough to project six vertices (out of eight) on to the image plane. The next step is to determine the pixels in the image plane which are covered by the projection footprint. The projection process is performed once for a cell and a template projection footprint is constructed. This projection template defines those pixels which are inside the hexagonal footprint. In order to project the other cells it is only necessary to project one vertex of that cell and then to match the template to the projection coordinates accordingly. There are different ways to construct the projection template, which are discussed by Wan et al [16]. The projection template needs to be reconstructed whenever the viewing point or direction is changed.

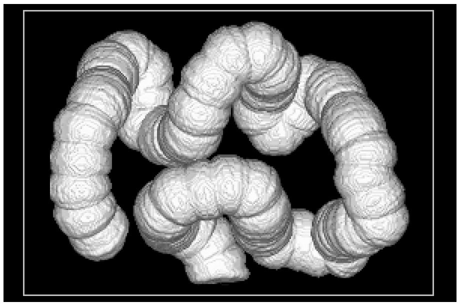

Figure 2. Synthetic colon data set used to test the method

2.2. The distance buffer

A distance 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 initialized to a value that indicates the distance is not yet calculated. During the ray casting process, as the image plane is swept pixel by pixel, the corresponding distance buffer entry 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 non-space cell. If the distance has not been previously calculated, then standard ray casting is applied. In this case cells along the ray are traversed while no non-space cell has been encountered. This can be performed efficiently using the space or non-space status determined in the pre-processing stage. When the first non-space cell is reached, this cell is projected on to the image plane. The distance buffer entries corresponding to the projection footprint, which have not been determined, will be updated subsequently.

2.3. Experimental Set-up

The presented method was implemented on an IBM compatible personal computer. A computer generated synthetic colon data set of 200 slices with 250x150 pixels per slice was used to test the method. A volume rendering visualisation of this data set is illustrated in figure 2 and is freely available from the authors. An animation based on the centre line of the colon was performed to compare the rendering speed of our accelerated method and the basic ray casting. We also validated the method by visualising and navigating interactively within parts of the colon using the visible human data set [17].

3. Results

Compared to basic ray casting our accelerated method was approximately 5 times faster when applied to the synthetic data set and 3.5 times faster when applied to the visible human data sets. Figure 3 illustrates a single frame of the visualisation of the inside of the visible female colon. Figure 3a shows the result of a basic ray casting. Figure 3b shows the same frame produced using the acceleration method and figure 3c shows only those pixels which did not benefit from space leaping. In figure 3c only 11074 out of 62500 pixels are presented, i.e. to produce the image via the accelerated method only 17.72% of rays are traversed through space and for the rest, ray casting started from non-space voxels.

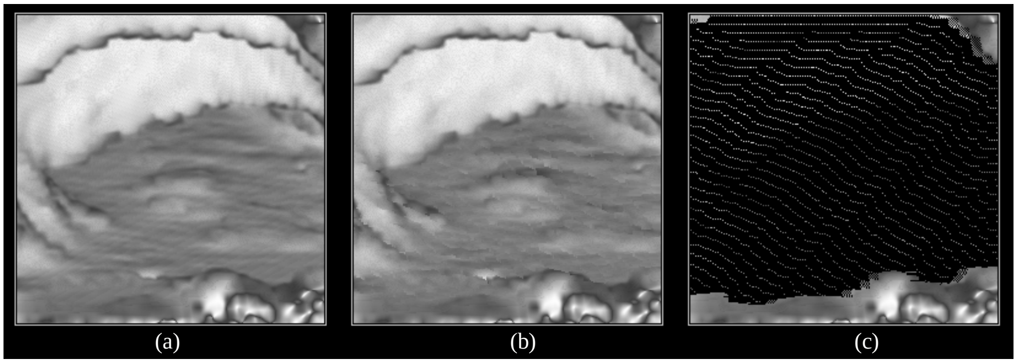

Figure 3. Visible female colon visualisation, (a) basic ray casting, (b) accelerated method, and (c) same as (b) but pixels which benefit from space leaping are excluded.

4. Discussion

In volume ray casting the spatial resolution of a volume data set is not necessarily the same as the resolution in the image plane. Let’s define the ratio of volume voxel size to the image plane pixel size as the enlargement factor. In virtual endoscopy where the viewpoint and image plane are positioned inside the volume data set, considering the resolutions of current CT scanners, the enlargement factor is greater than one. In a virtual colonoscopy application, with typically 512x512 pixels per slice and a 1mm slice interval, the enlargement factor to produce a 256x256 image is approximately 7. The amount of acceleration achievable using the presented method depends on the enlargement factor. A higher enlargement factor results in a larger area covered by projecting cells on to the image plane. Therefore, the overhead of calculation required to traverse the space for a ray and to project a cell is divided between more rays. Thus less space traversing and cell projecting is required to produce an image, and thereby a higher frame rate can be delivered. In a virtual colonoscopy application with the parameters mentioned above, each cell projection covers approximately 20 pixels. Similar rates of acceleration apply to virtual bronchoscopy, blood vessels visualisation, and other virtual endoscopy applications.

A small difference between the images produced by this method and the basic ray casting can occur due to the approximation used to select the group of rays that reaches the same non-space cell. Figure 4 illustrates the cause of the difference. Consider a ray A, the first non-space cell is specified as C$_1$. The corridor of rays corresponding to the projection of C$_1$ lies between the two dashed lines. The rays close to the upper dashed line reach another non-space cell C$_2$ before C$_1$. This situation arises when the first ray is close to the object boundary. The bypassed cells (e.g. C$_2$) are always boundary cells, which are not opaque and are neighbours to space cells.

In applications where zero degradation is required our method can still be useful as a pre-viewer allowing the user to navigate inside the object and the basic ray casting can be applied as soon as the viewpoint is fixed.

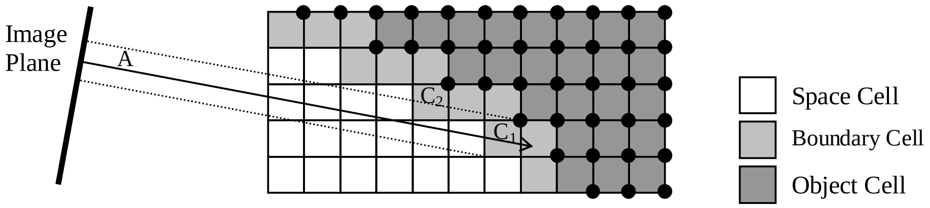

Figure 4. Possible differences caused by the projection approximation

References

1. Cancer Facts and Figures 2002, The American Cancer Society, http://www.cancer.org, last visited May 2002.
2. Z. Kmietowicz. “NHS aims to improve care for colorectal cancer”, *BMJ* **315**, pp. 1485-1486, 1997.
3. D. Vergano. “Annual Colorectal Cancer Screening Cuts Death Risk”, *Medical Tribune* **40(7)**, pp. 19, 1999.
4. P. Pescatore, T. Clucker, J. Delarive, et al “Diagnostic Accuracy and Interobserver Agreement of CT Colonography (Virtual Colonoscopy)”, *Gut* **47**, pp. 126-130, 2000.
5. D.K. Rex, C.S. Cutler, G.T. Lemmel, et al “Colonoscopic Miss Rates of Adenomas Determined by Back-to-Back Colonoscopies”, *Gastroenterology* **112**, pp. 24-28, 1997.
6. D.J. Vining, D.W. Gelfand, R. Bechold, et al “Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality”, In *Proceedings of Annual Meeting of the American Roentgen Ray Society*, pp. 014, 1994.
7. R. Shahidi. “Surface Rendering Versus Volume Rendering in Medical Imaging: Techniques and Applications”, *In Proceedings of Visualization ’96*, pp. 439-440, 1996.
8. R. Yagel & Z. Shi “Accelerating Volume Animation by Space-Leaping”, *In Proceedings of IEEE Visualization’93*, pp. 62-69, 1993.
9. R. Yagel. “Towards Real Time Volume Rendering”, *In Proceedings of GRAPHICON'96*, Vol. 1, pp. 230-241, 1996.
10. J. Danskin & P. Hanrahan “Fast algorithms for volume ray tracing”, *Workshop on Volume Visualization*, pp. 91-98, 1992.
11. M. Sharghi & I.W. Ricketts “A Novel Method for Accelerating the Visualisation Process Used in Virtual Colonoscopy”, In *proceedings of 5th International Conference on Information Visualisation*, pp. 167-172, IEEE Computer Society, 2001.
12. S. You, L. Hong, K. Junyaprasert, et al “Interactive Volume Rendering for Virtual Colonoscopy”, In *Proceedings of Visualization ’97*, pp. 433-436, 1997.
13. R. Avila, L.M. Sobierajaski, & A. Kaufman “Towards a Comprehensive Volume Visualization System”, In *Proceedings of Visualization’92*, pp. 13-20, 1992.
14. K. Zuiderveld, A.H.J. Koning, & M.A. Viergever “Acceleration of Ray Casting Using 3D Distance Transform”, In *Proceedings of Visualization in Biomedical Computing*, pp. 324-335, 1992.
15. M. Wan, Q. Tang, A. Kaufman, Z. Liang, et al “Volume Rendering Based Interactive Navigation Within the Human Colon”, In *Proceedings of IEEE Visualization*, pp. 397-400, 1999.
16. M. Wan, S. Bryson, & A. Kaufman “Boundary Cell-Based Acceleration for Volume Ray Casting”, *Computers & Graphics* **22(6)**, pp. 715-721, 1998.
17. U.S. National Library of Medicine, The Visual Human Project, http://www.nlm.nih.gov/research/visible.