Fast and Flexible Iso-Surfacing for Virtual Endoscopy

André Neubauer, Lukas Mroz, Helwig Hauser, Rainer Wegenkittl
neubauer@vrvis.at, lukas.mroz@tiani.com, hauser@vrvis.at, wegenkittl@vrvis.at

VRVis Research Center
Vienna / Austria

1 Abstract

Virtual endoscopy is an important tool for physicists, that can help them planning a real endoscopic examination or serve as a navigation aid during a physical endoscopy process. One of the strengths of virtual endoscopy is that not only the investigated organ can be displayed, but also surrounding objects like bones, blood vessels, or a tumor. This paper introduces a new visualization algorithm, which performs well visualizing the surface of the investigated organ, as well as outer structures, while keeping image quality and flexibility at a high level. Also, the preprocessing phase, in which objects of interest are chosen and segmented, is covered.

2 Introduction

Endoscopy is an important technique for medical diagnosis. Endoscopes, small cameras, are inserted into patients via natural holes or small artificial incisions to examine hollow organs or cavities within the body. To perform surgery or take tissue probes, a second device, such as tiny scissors or forceps, can be inserted along with the endoscope. Since endoscopic procedures are associated with discomfort for the patients, it is important to keep them as short as possible by working efficiently. Virtual endoscopy is a relatively new method of diagnosis, which provides simulated visualizations of inner organs similar or equivalent to those produced by standard endoscopic procedures and can support the endoscopist in many ways. It is performed by placing a virtual camera into a three dimensional data volume which most commonly is acquired from computer tomography (CT) or magnetic resonance imaging (MRI). Data acquisition is therefore non invasive and causes almost no discomfort for the patient. Another advantage of virtual over physical endoscopy is the extended field of view: the virtual endoscope can easily be adjusted to point in any direction. Furthermore, the field of view needs not be confined by organ walls, instead, also external structures can be viewed. If no physical interference is required at all, virtual endoscopy might even in some cases replace the real endoscopic procedure. Otherwise, it is still a powerful tool which can aid the endoscopist to identify regions of interest within the investigated organ prior

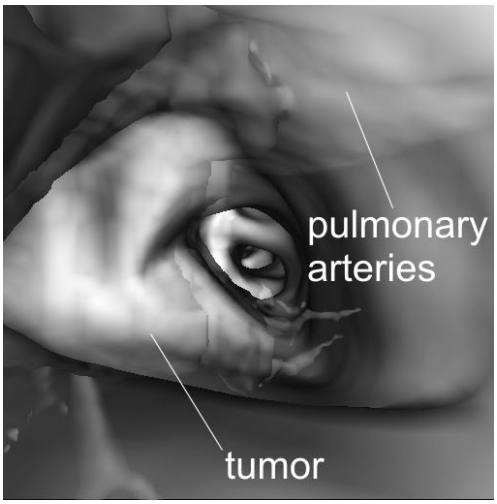

Figure 1: snap shot from a virtual endoscopy investigation inside the trachea

to a real endoscopic examination or to navigate the physical endoscope through a widely branched organ.

An interesting application of endoscopy is *trans-bronchial biopsy*. If a tumor is detected inside a patient's lung, a device has to be inserted into the body to take a biopsy from tumor tissue, which is then tested for malignity. Tumors are often hidden behind bronchial walls and therefore cannot be seen through a camera inserted into the bronchial tree. Also, bronchial walls occlude the aorta and other important blood vessels that must not take any damage during the biopsy. A virtual camera, however, can display the bronchial walls semi-transparently and thus guide the physicist to the tumor and help them find the correct position for biopsy. In this application, virtual endoscopy can be carried out either prior to the real endoscopic procedure as an exercise for the endoscopist, or even during the physical examination as a navigation aid. There is no doubt that especially in the latter case, virtual endoscopy must not induce any artificial delays. The process of rendering the virtual camera image should be fast. An elaborate visualization technique, which displays the surface of the organ under investigation semi-transparently and specified objects of interest (here: the tumor and the vessel tree) opaquely (see figure 1) within reasonable frame times, is needed.

Cell-based first-hit ray casting, a new volume visualization technique suitable for a large variety of virtual endoscopy scenarios, is presented and evaluated with respect to the trans-bronchial biopsy application in this paper.

3 Related Work

For the virtual endoscopy application, visualization methods based on iso-surface rendering are more widely used than those based on volume rendering, since volume rendering techniques usually lack either speed or visual quality. An iso-surface is a

surface of constant value (the so-called *threshold*) inside a three-dimensional data volume. There are two types of iso-surfacing techniques.

The first are *surface fitting methods* (SFM) which first convert volumetric data into a mesh of polygons using a surface extraction algorithm, e.g., the *marching cubes* technique [12]. The mesh representing the iso-surface is subsequently rendered using dedicated polygon rendering hardware. Due to hardware acceleration, this results in fast rendering, if the number of polygons is not too large. The most important disadvantage is limited flexibility. The iso-surface extraction procedure must be carried out every time the threshold (and therefore also the iso-surface) is changed. However, the provision of means of interactive threshold modification is important to allow the physicist to adjust the visual appearance to their needs, which prohibits the use of traditional SFM. During the past few years, some surface fitting techniques that execute a surface extraction procedure during each frame have been introduced. However, those methods either do not yet perform convincingly well or provide functionality that is limited in one way or the other. The fast technique proposed by Hietala et al. [7], for example, does not allow for interactive threshold modification.

The second type of iso-surfacing techniques is first-hit ray casting. Like in the traditional ray casting algorithm [9], each pixel is associated with a viewing ray. However, pixel colors are not accumulated by taking evenly spaced samples along the rays, but determined by intersecting the rays with the predefined iso-surfaces. Pure first-hit ray casting is in general too slow to be applicable in the virtual endoscopy scenario. Many ideas have been presented to accelerate first-hit ray casting in particular and ray casting in general. Important categories of techniques that optimize first-hit ray casting without altering image quality are:

acceleration of empty space traversal Ray sampling intervals are chosen according to the probability of an iso-surface being intersected [3, 6, 16].

data set reduction : interesting parts of the data set are identified prior to the rendering process. Rendering is then confined to those regions [11, 15].

improvement of caching behavior Thrashing is minimized by maximally utilizing cached data before they are replaced [8, 14].

Cell-based first-hit ray casting, the new iso-surfacing algorithm that will be introduced in the next section, utilizes each of these three concepts to optimize rendering performance while keeping image quality at a high level.

Also, some acceleration techniques for first-hit ray casting, which trade image quality for speed, have been proposed:

simulation of perspective projection Rays diverge in a perspective fashion, but are locally orthogonal and can be accelerated using templates [8, 14].

usage of frame-to-frame coherency The result image of a frame is used to predict the next frame. The prediction is then refined [18].

usage of pixel-space coherency Only a subset of the pixel colors is determined using ray casting, the remaining ones are interpolated [10].

Another way to tackle the problem of interactive visualization for virtual endoscopy is image-based rendering. Wegenkittl et al. [17] introduced a method for the trans-bronchial biopsy application. Virtual cubes are centered on discrete points along a predefined path. In a preprocessing phase, an image is rendered to each of the six sides of each virtual cube, using direct volume rendering. During the interaction phase, it is possible to rotate the camera to any direction by accordingly projecting the six sides of the cube associated with the current view point to the screen. Aided by hardware acceleration, this technique achieves highly interactive frame rates. However, it trades flexibility for speed. Not only must the camera move on the predefined path, also the visual appearance of investigated structures cannot be adjusted at interaction time, since the selection of transfer functions and visualized objects must be done before the rather time-consuming preprocessing phase.

4 Segmentation

In order to provide the possibility of displaying objects of interest, like the aorta, the pulmonary vessels or a tumor, opaquely behind the semi-transparent organ surface, the data set must be segmented in a preprocessing step. Roughly, two types of objects can be distinguished with respect to their representations in data sets obtained from standard modalities like CT and MRI:

Objects of the first type are represented in the volume by voxel data values which are on a higher level than the surroundings of the objects. Examples are bones and blood vessels (if flooded by a contrast agent) in a CT image. Those objects have the advantageous property that their surfaces approximately coincide with properly chosen iso-surfaces. They can be segmented by a simple thresholding-based segmentation technique: An initial threshold is chosen. Neighboring voxels whose data values are above this threshold are merged to objects. The number of objects can then arbitrarily be set to n by deleting all but the n biggest objects. Then the threshold is iteratively decreased and each object is enlarged to include neighboring voxels above the new threshold. Since each voxel may only belong to one object, the objects will, when the threshold is small enough, collide and cease growing. Now, each object can be rendered independently with an individual threshold adapted to the viewer's needs. Figure 2 shows some objects representing bones as well as an object that includes the aorta and, if the threshold is decreased, also other major blood vessels and the kidneys. The appearance of this object can be interactively altered without changing the appearance of any other object.

The majority of objects (e.g., almost all inner organs) belong to the other group. They require more sophisticated treatment, because they have two disadvantages:

The first disadvantage is that iso-surfaces cannot be used to represent object boundaries, since the representations of those objects in the data volume can be

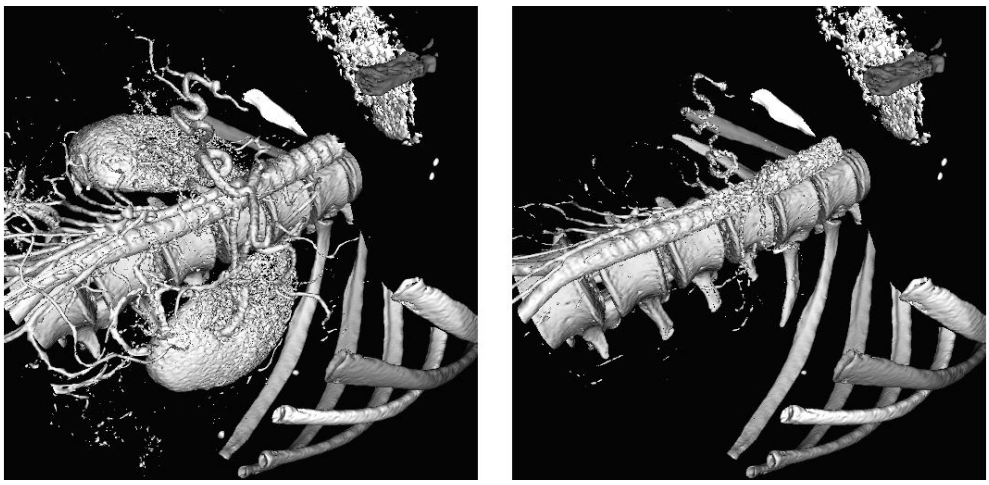

Figure 2: altering the threshold of one object

bounded by regions of lower as well as regions of higher data levels. The segmentation process must take into account gradient information. The *watershed* algorithm [2, 4] is a well known technique that is suitable for this task. This technique, when used to segment a two-dimensional image, takes the gradient image as an input height field which is virtually flooded with water. This process produces *catchment basins* which correspond to objects. For the segmentation of a volume, an analogous three-dimensional procedure is applied.

The second disadvantage is that sometimes parts of object boundaries are not indicated by significant gradients, because tissues neighboring the object of interest can also be represented by data values on the same or a very similar data level as the object. Here, the segmentation process must be aided by manually separating objects. An algorithm called *watershed-from-markers* [5] can be applied. It is a variation of the watershed algorithm, which takes a manually placed set of markers into account.

After such an object is segmented, the problem of visualization remains. Since the object boundary differs strongly from every possible iso-surface, an artificial iso-surface must be generated. One possible way to do this is to lift the complete object representation to a higher level, above the object surroundings, by adding an offset to all voxel data values belonging to the object representation before the rendering process. Now, the object can be rendered using iso-surfacing. The appearance of the artificial iso-surface is, however, very much dependent on the offset used. Low offset values, which lift the object just slightly above its surroundings, leave the iso-surface very threshold-dependent. This may be undesirable, as for example in the trans-bronchial biopsy case, where the lung tumor should appear in its real size for a wide range of different thresholds. Large offset values, on the other hand, cause high spatial frequencies in the volume and therefore terracing: sloped surfaces degenerate to sequences of close to flat planes. The lung tumor, in this case, would appear as what its representation, obtained from the modality, is: a finite collection of discrete samples, rather than a natural formation.

5 Cell-Based First-Hit Ray Casting

Cell-based first-hit ray casting is a new fast first-hit ray casting algorithm, usable for virtual endoscopy, which renders a number of predefined iso-surfaces. This section gives a detailed description of the algorithm.

The algorithm is based on the organization of volume data as a min-max octree. Each node of the octree represents a cubic sub volume and stores the minimum and maximum data values of this sub volume. Leaf nodes of the octree represent so-called macro-cells which are cubic sub volumes consisting of n^3 data cells ($(n+1)^3$ voxels). Usual values of n range between 3 and 10. By traversing the octree, macro-cells containing a part of one of the iso-surfaces can be found efficiently. Each of those macro-cells is trimmed to the smallest cuboid containing the macro cell's portions of the iso-surfaces and then projected to the screen plane. The projection is rasterized and a ray is cast through each covered pixel. This *local ray* is only tracked through the currently processed macro-cell. Through the application of a fast voxel traversal algorithm [1], cells, which are pierced by the ray and contain a part of an iso-surface, are detected. Within these *surface cells*, a ray-surface intersection test is performed. If an intersection is found, the gradient vector at the intersection point is calculated and a color is computed for the corresponding pixel, and, if the iso-surface is opaque, the pixel is marked such that no more local rays are cast at this pixel during the current frame.

It is easily understood that macro-cells must be processed in viewing order to avoid visibility artefacts. Therefore, during octree traversal, child nodes of the currently processed node must be sorted according to the viewing vector and accessed in the correct order.

5.1 Rasterization

After a macro-cell has been projected to the image plane, a rasterization process determines the pixels, at which local rays have to be cast through the macro-cell. Faces of macro-cells, which would be visible from the current camera position, if the macro-cell was an entirely opaque cube in space, are called *visible*. At most 3 macro-cell faces are visible, depending on the relative position of the macro-cell with respect to the view point. Analysis of the vector from the view point to the center point of the macro-cell leads to three *potentially visible* macro-cell faces. Each of them must be tested for real visibility. The vertex, that belongs to all potentially visible faces, is called *center vertex*.

The outline of a macro-cell projection is defined by the projections of the vertices of all visible macro-cell faces. It is made up by either 4 (one macro-cell face is visible) or 6 (two or three macro-cell faces are visible) projections of vertices. These projections will be referred to as *silhouette points*. Each vertex is assigned an *order*, which is the number of visible macro-cell faces, that the vertex is part of. The projection of a vertex v is a silhouette point, if and only if the vertex v has an order of one or two.

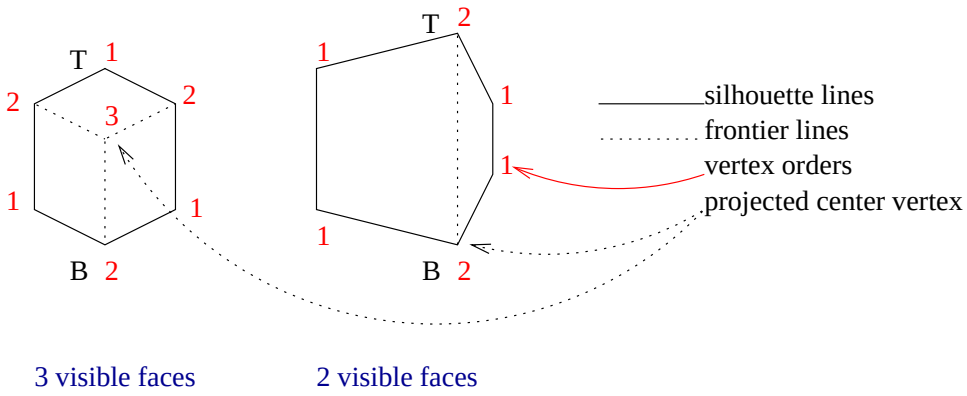

Figure 3: possible projection outlines

Silhouette lines connect silhouette points to form the boundaries of the macro-cell projection. The projection of the center vertex is connected to the projections of all other vertices, which have an order that is greater than one, by *frontier lines*. Frontier lines mark the boundaries between the projections of visible macro-cell faces. Figure 3 shows two examples of macro-cell projections.

The rasterization process moves from the top (T) to the bottom (B) silhouette pixel of the projection. Rasterized silhouette lines define a left and a right bound pixel for each scan line. For each pixel in between, a local ray is cast. The position of each pixel relatively to intersections of the scan line with rasterized frontier lines indicates, through which macro-cell face the local ray enters the macro-cell. Thus, ray entry points, the starting points of local rays, can be found efficiently by simply intersecting each ray with the plane of its entry face.

5.2 Early Scan Line Termination

Early scan line termination (EST) is an optimization technique that reduces the number of local rays. As described above, each scan line intersected by the projection of the currently processed macro-cell is rasterized from the left to the right projection boundary. EST stops the rasterization process of a scan line as soon as one pixel remains blank because its local ray through the macro-cell did not intersect an iso-surface. It is easily understood that EST cannot be applied every time a local ray leaves a pixel blank, as this would sometimes lead to ray-surface intersections being missed and induce clearly visible errors. So, before rasterization, those parts of the macro-cell projection, in which EST can be safely applied, must be identified. Surface cells, within which ray-surface intersections might be missed due to EST, prohibit EST. They are detected and projected to the image plane in order to identify a *critical region*. EST can only be executed outside the critical region. Figure 4 shows two examples of iso-surfaces inside macro-cells. Curves and lines connecting those pixels, at which EST takes place, are labelled "EST" and drawn boldly in the figure. In the image on the left, all scan lines can be terminated early, because there is no critical region. In the example on the right hand side, the application of EST is confined by a critical region.

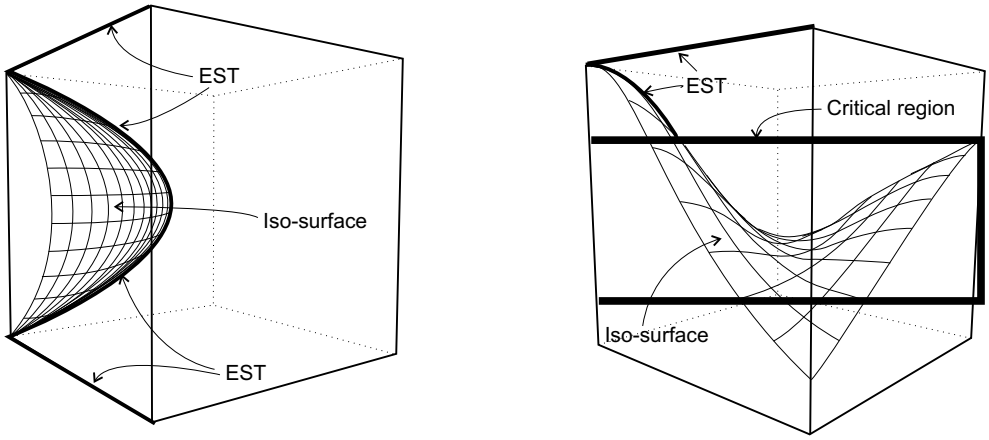

Figure 4: two macro-cells and the applicability of early scan line termination (EST) if scan lines are processed from left to right

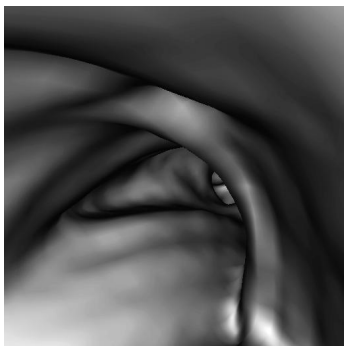

Figure 5: a snap shot from a virtual flight through a trachea

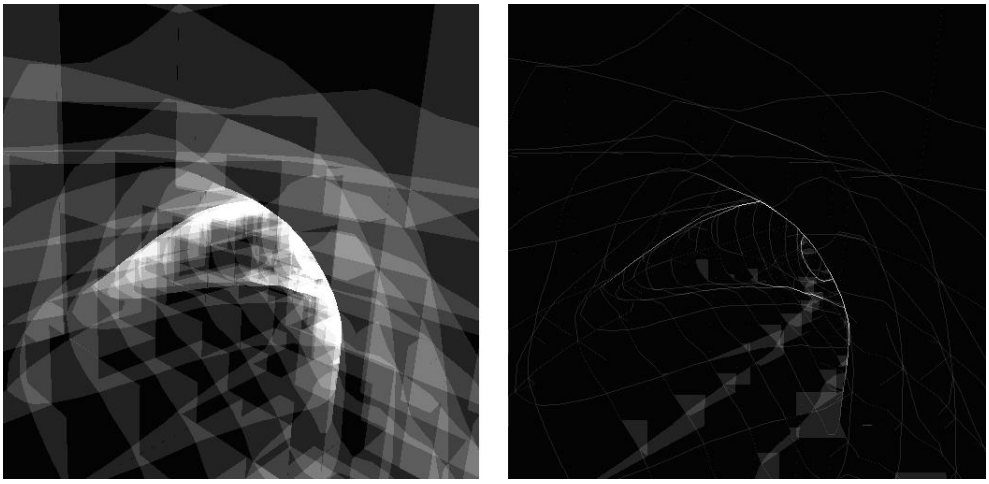

Figure 6: numbers of local rays per pixel without (left) and with (right) the usage of early scan line termination

To make the concept of early scan line termination more effective, the rasterization algorithm can be adapted according to the geometry of the iso-surfaces inside the macro-cell to allow also scan lines being processed from right to left, or treating pixel columns as scan lines and processing them from top to bottom or bottom to top. If, for example, scan lines are processed from bottom to top in the example on the right hand side in figure 4, all scan lines can be terminated early.

EST does a good job reducing the numbers of local rays per pixel. Figure 5 shows a snap shot from a virtual flight through a human trachea. Figure 6 depicts the numbers of local rays per pixel needed for rendering this image using cell-based first-hit ray casting both with and without EST. The brighter the color of a pixel, the more local rays were cast at that pixel. The wire-frame-like appearance of the image on the right is due to the fact that local rays, which initiate EST, leave the pixel blank. Thus, at pixels at which EST is performed at least one more local ray must be cast. In this example, EST reduces the average number of local rays per pixel from 2.803 to 1.143, which saves, for an image resolution of 512×512, about one second of rendering time for the frame.

The decision, as to whether a surface cell can be considered unsafe and its projection should be included in the critical region, is based on calculating a representative normal vector for the iso-surface inside the surface cell and testing its direction in screen space. Since a representative surface normal might describe a part of an iso-surface too inaccurately, critical parts of the iso-surface might fail to be detected. Thus, EST can still lead to parts of iso-surfaces being missed. This results in holes in the iso-surface which can be detected after the regular computation of a frame, and filled. The exact criteria for surface cells being considered critical and the hole recovery algorithm are described in the master's thesis about cell-based first-hit ray casting [13].

6 Results

The algorithm described was implemented and timed on an Intel P4 1900 MHz machine and assessed in the trans-bronchial biopsy application using a $364 \times 264 \times 144$ data set of a human chest. The important background objects (aorta, pulmonary arteries and tumor) were pre-segmented. For the tumor, a watershed-from-markers algorithm was used, the vessels were segmented using the thresholding-based technique (see section 4). Three sequences, each consisting of 100 images (512×512 pixels), were rendered and timed: one sequence displaying only the surface of the trachea, one displaying only the background objects, and one displaying the surface semi-transparently over the background objects. Figure 7 shows two snap shots out of each sequence, table 1 gives timings.

Cell-based first-hit ray casting performs similarly to conventional first-hit ray casting (where rays are tracked from the eye point to the intersection points with iso-surfaces) in the surface-only scenario. Conventional ray casting, however, is outperformed quite significantly, if background objects are rendered. The main reason for the good performance are the short paths that rays have to traverse.

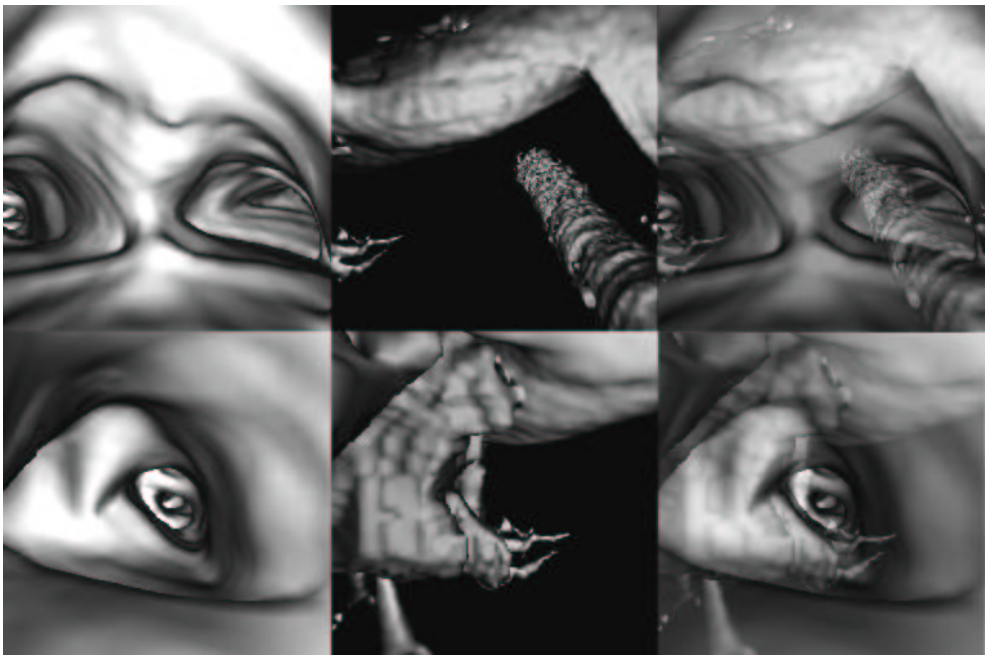

Figure 7: three image sequences

algorithm used	minimum	maximum	average
surface only	0.873	1.249	0.968
background objects only	1.224	1.769	1.502
surface and background objects	2.186	2.896	2.532

Table 1: frame times in seconds for sequences of images

Average numbers of local rays per pixel range between 1.1 and 1.4 for surface only and background only and between 2.5 and 3 for the semi-transparent views. Performance is heavily influenced by the number of objects and the number of different thresholds, since these determine, how many macro-cells have to be processed. Also, the macro-cell size has an influence on rendering speed, as it determines the number of macro-cells, the length of local rays and the efficiency of early scan line termination.

7 Conclusion and Future Work

This paper has dealt with fast and flexible visualization for virtual endoscopy without the use of hardware acceleration. It has discussed the segmentation of important objects, which serve as navigation aid in the endoscopy process, as well as the rendering process. Cell-based first-hit ray casting, a new visualization technique, which is well applicable, for a large spectrum of scenarios in virtual endoscopy, has been introduced. It performs well visualizing the inner surfaces of the investigated organ, as well as surrounding objects. As expected, rendering speed is significantly below that of a hardware-accelerated image-based technique, however, huge gains in flexibility easily make up for that, since rendering speed is still at an acceptable level. Some ideas exist to further improve the performance of cell-based first-hit ray casting. They include, among others, finding an optimal trade-off between effectiveness and security of early scan line termination and improvement of the used data structures to reduce per-macro-cell overhead.

8 Acknowledgements

This work has been done as part of the basic research on visualization (http://www.VRVis.at/vis/) at the VRVis research center in Vienna, Austria (http://www.VRVis.at). Thanks to Tiani Medgraph (http://www.tiani.com) for providing medical data sets.

References

[1] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. In *Proc. of Eurographics '87*, pages 3–10, 1987.

[2] M. Bruckschwaiger. Evaluation of binary segmentation techniques regarding their usability in medical volume visualization. Master's thesis, Institute of Computer Graphics and Algorithms, Vienna University of Technology, 2002.

[3] D. Cohen and Z. Shefer. Proximity clouds - an accelerated technique for 3D grid - traversal. *The Visual Computer*, 11:27–38, 1994.

[4] M. Couprie and Gilles Bertrand. Topology grayscale watershed transformation. In *Proc. SPIE Vision Geometry V*, volume 3168, pages 136–146, 1997.

[5] P. Felkel, M. Bruckschwaiger, and R. Wegenkittl. Implementation and complexity of the watershed-from-markers algorithm computed as a minimal cost forest. In *Proc. of EUROGRAPHICS 2001*, 2001. Computer Graphics Forum 20(3).

[6] J. Freund and K. Sloan. Accelerated volume rendering using homogeneous region encoding. In *Proc. of IEEE Visualization '97*, pages 191–196, 1997.

[7] R. Hietala and J. Oikarinen. A visibility determination algorithm for interactive virtual endoscopy. In *Proc. of IEEE Visualization 2000*, pages 29–36, 2000.

[8] A. Law and R. Yagel. Multi-frame thrashless ray casting with advancing ray-front. In *Proc. of Graphics Interfaces 1996*, pages 70–77, May 1996.

[9] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(5):29–37, 1988.

[10] M. Levoy. Volume rendering by adaptive refinement. *The Visual Computer*, 6(1):2–7, 1990.

[11] W. Li, M. Wan, B. Chen, and A. Kaufman. Virtual colonoscopy powered by volumepro. Indexed at `http://www.cs.sunysb.edu/`.

[12] W. E. Lorensen and H. E. Cline. Marching cubes: a high resolution 3d surface construction algorithm. In *Proc. of SIGGRAPH'87*, pages 163–169, 1987.

[13] A. Neubauer. Cell-based first-hit ray casting. Master's thesis, VRVis Research Center, 2001. Indexed at http://www.VRVis.at/.

[14] K. L. Novins, F. X. Sillion, and D. P. Greenberg. An efficient method for volume rendering using perspective projection. *Computer Graphics*, 24(5):95–100, November 1990.

[15] H. Shen, C. D. Hansen, Y. Livnat, and C. R. Johnson. Isosurfacing in span space with utmost efficiency. In *Proc. of IEEE Visualization '96*, pages 287–294, 1996.

[16] B. T. Stander and J. C. Hart. A Lipschitz method for accelerated volume rendering. In *Proc. of IEEE Visualization '95*, pages 107–114, 1995.

[17] R. Wegenkittl, A. Vilanova, B. Hegedüs, D. Wagner, M. C. Freund, and E. M. Gröller. Mastering interactive virtual bronchioscopy on a low-end pc. In *Proc. of IEEE Visualization 2000*, pages 461–464, 2000.

[18] R. Yagel and Z. Shi. Accelerating volume animation by space leaping. In *Proc. of IEEE Visualization '93*, pages 62–69, 1993.