

# VirEn: A Virtual Endoscopy System

Anna Vilanova  
Andreas König  
Eduard Gröller \*

## Abstract

Virtual endoscopy systems are promising tools for the simplification of daily clinical procedures. In this paper, a conceptual framework for a virtual endoscopy system (VirEn) is proposed, which is intended to be a highly interactive system. Research efforts have concentrated on the generation of an optimal path for the automated navigation of the data set. Extensions to existing thinning algorithms used to generate the optimal path are presented and discussed. First results produced with VirEn are shown.

**Keywords:** Volume Visualization, Virtual Endoscopy, Navigation, Thinning

## 1 Introduction

In recent years, the importance of medical imaging in patients care systems is increasing due to the evolution of 3D imaging modalities, like CT or MRI. Computer graphics techniques for the visualization of the acquired 3D imaging data sets are topics of research. Virtual endoscopy is a promising new technique to explore hollow organs and anatomical cavities. The fields of application are manifold:

- Non-invasive diagnostic endoscopy (e.g. colonoscopy) avoids the risks associated with real endoscopy.
- Virtual endoscopy can be used for educational purposes like endoscopists training (e.g. sinus surgery).
- Special parts of the human body, which are impossible to access with a real endoscope (e.g. blood vessels, thoracic aorta).
- A special field of application is the usage of virtual endoscopy for surgery planning.

Several virtual endoscopy systems have been proposed in recent years. In spite of the fact that all of these systems are constrained to different applications, they do have similar components.

This paper presents a proposal for the development of an interactive virtual endoscopy system. Preliminary results produced with a test implementation of this framework will be shown. Single components of the system,

which have been the matter of emphasized research investigation, will be described in detail. The generation of an optimal path through hollow organs is the main topic for original research results.

Following this introduction, the Virtual Endoscopy system (VirEn) and its components are described. In section 3 the related work for the different components of the Virtual Endoscopy system is discussed. So far, our efforts have concentrated on some scientific elements of the whole system. One of them is the optimal path calculation, which will be described in detail in section 4. In section 5, a brief description of the implementation is given. Finally, conclusions and future work will be discussed in section 6.

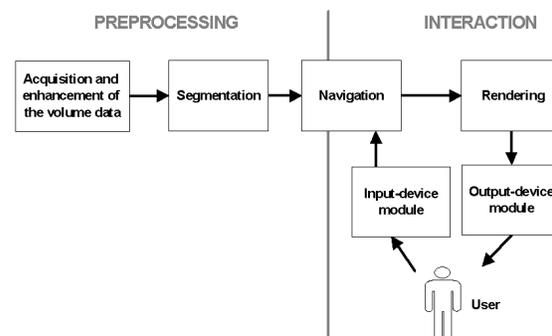


Figure 1: VirEn: System overview.

## 2 VirEn: Structure of a Virtual Endoscopy system

VirEn is an interactive Virtual Endoscopy system. The main idea of VirEn is that having the volume data as input of the system, after some preprocessing the user can explore the data in a way similar to a real endoscopy or, at least, obtain similar information. In this section, we will explain the structure and elements of this system. A global structure of VirEn is shown in figure 1. The system basically consists of preprocessing and interaction modules. The modules in the preprocessing group are less time critical since they are calculated just once for every input data set. Computations carried out by the interaction modules are done on the fly during user navigation and therefore

\* {anna,koenig,groeller}@cg.tuwien.ac.at

execution speed is crucial.

The virtual endoscopy system consists of the following main elements (see figure 1):

**Acquisition and enhancement of the volume data sets.**

This component provides the initial input to the system, which is a 3D image data set. The data is usually acquired by one of the usual different medical imaging modalities (CT, MRI, etc). Frequently, there is the need to apply image processing techniques (e.g., noise removal) to the 3D image in order to increase the quality of the data and facilitate the following processing steps. The resulting output of this module is an enhanced volume data set.

**Segmentation** separates an object from its surroundings in a 3D imaging data set. The result of this process can be data of the same dimension, a subvolume (3D), or data of lower dimensions, such as surfaces (2D), border lines (1D), points (0D) and so on. The simpler the data is, the easier and faster it can be viewed, but also the more information is lost and the interpretation of the data is also more difficult.

The segmentation module is used for the definition of the object in the whole data volume which the user is interested in (e.g., colon, trachea).

**Navigation.** The navigation module is concerned with the on-the-fly inspection of narrow tubular structures typical in endoscopy. It includes the interaction of the user to control camera movement, and deals with mapping the input device movements to camera parameter modifications. The user should get neither a “lost-in-space” feeling because of dealing with too many parameters nor a frustration feeling due to a heavily constrained navigation environment.

A near real-time frame rate is also an important goal for a feasible navigation. Imaging devices are producing increasingly better quality volume data sets, but at the expense of high storage requirements. In virtual endoscopy, the viewpoint is moving inside the data set and just a little part of the whole data will be seen. The navigation component is taken advantage of this fact to reduce the data that will be sent to the rendering module. The navigation module provides the data for the rendering module and also the camera parameters like camera position and orientation.

**Input-device module.** This module is handling the hardware devices used in the interaction with the user. There are many choices that can be used for the interaction with a virtual endoscopy systems. Different classes of devices ranging from common desktop hardware (keyboard and 2D mouse) over 3D devices with 6 degrees of freedom and haptic feedback, to a special virtual endoscope device [12].

**Rendering.** Once the camera position is determined and the data is prepared, rendering has to be performed.

There are several techniques that can be applied to render the data set (which does not necessarily have to be volume data any more because of the segmentation step) depending on the accuracy and the desired frame rate. It is important to point out that rendering should use perspective projection since the viewpoint is inside the data set and depth information is lost using parallel projection. Stereo viewing can also be applied in order to enhance the realistic impression.

**Output-device module.** This module handles the devices used to present the user the result of the rendering module. As for the input-device module, there are several choices (e.g., monitors, head mounted displays, etc).

We can distinguish two main issues in virtual endoscopy: *accuracy* and *user interaction*. Virtual endoscopy can be used in applications like diagnosis and surgical planning and therefore the data must be accurate enough not to lead the physicians to a wrong decision. The accuracy problem concerns acquisition, segmentation and rendering. Another main topic is *user interaction*. In order to get clinical acceptance, the user must be able to deal easily and fast with the system. This concerns the navigation and the rendering components.

In the presented paper, the authors focus on the segmentation, rendering and navigation modules. In the remaining part of this section, they will be described in further detail.

## 2.1 Segmentation

Segmentation is a research topic in many areas (e.g., computer vision, pattern recognition, image processing). There is a high degree of complexity in automating the segmentation process. Furthermore, a general, fully automatic segmentation tool is very difficult to achieve. Usually the degree of automatization of a segmentation procedure relates inversely proportional to the accuracy of the result. On the other hand a manual only segmentation is tedious and time-consuming.

Therefore most of the general segmentation techniques are semiautomatic. The user introduces some clues (threshold, a seed point [42], contour guiding [16, 27], and others) in order to guide the algorithm, which automatically calculates a part of the segmentation. These methods are a compromise between accuracy due to user supervision and being not as time consuming as a manual segmentation. The data is segmented in the VirEn system using one of the previous cited techniques depending on the type of organ which is going to be inspected.

## 2.2 Rendering

Volume rendering techniques are divided into two groups depending on whether they use a previous segmentation step to reduce the data or not.

One group of techniques are the so-called direct volume

rendering techniques (e.g., ray casting [19] [33] [34], splatting [36]). They directly visualize the volume data and no loss of information has to be taken into account. These techniques are rather computationally expensive. Although several acceleration methods (software [1] [4] [6] [18] [37] and hardware [5][30]) can be used, currently the achieved frame rates are not typically real-time capable without compromising the quality of the results.

Another group of techniques are surface rendering methods. These techniques rely on a previous segmentation (section 2.1) that produces as a result the surface of the object of interest. This surface can be represented as a mesh of polygons. Therefore, rendering can be done with the usual graphics hardware support. It is obvious that accuracy is lost with this approach, since the interior part of the object does not contribute to the visualization. The shape of the surface but not its surroundings can be inspected. These techniques are useless in some diagnostic cases where not just the shape, but, also the surrounding tissue is important (e.g., trying to decide whether a polyp is benign or not).

Furthermore, segmentation is sometimes nearly impossible to achieve in a correct way, even manually. The segmentation step may introduce artifacts that might result in wrong visualizations [15] [21].

Segmentation techniques that generate data with lower dimension than a surface, like contour lines, could also be used. Along with the dimension, the accuracy is severely reduced in these cases.

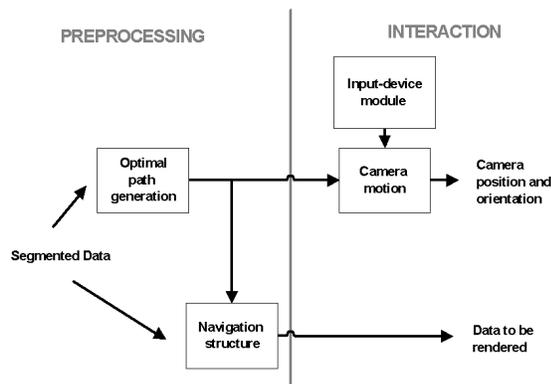


Figure 2: VirEn: Navigation module

## 2.3 Navigation

The navigation module primarily deals with 3D interaction and with the generation of data structures that can accelerate the rendering process, using the locality of the viewing frustum within the spatial extend of the data set (see figure 2). Navigation in virtual endoscopy has some peculiarities that have to be mentioned:

- In general, the physicians want to inspect the internal part of an organ, so they are not interested in going through its walls. Nevertheless, special applications like angiography of brain vessels might make this option desirable.
- The user would like to have a wide angle view during navigation, which can be achieved by keeping the camera as close to the center of the hollow organ as possible. When the camera position is moving, the camera will usually be looking to the end of the tunnel and not directly to the walls.

If an approximation of the optimal path can be predicted, it could be used for automated navigation or to improve the interactivity with the user.

As is shown in figure 2, the navigation module has been divided into different components. In the following, these components will be described.

**Optimal path generation.** The optimal path has to be connected and reside as close to the center of the hollow object as possible. In order to obtain this path, several approaches (from manual specification to automatic path generation) can be used.

An automatic way to generate this path is to calculate the so-called *skeleton*. One definition of a skeleton is: the locus of points that are geometrically centered with respect to the object boundary.

Usually, methods for the generation of the optimal path require a segmentation step first, since the input to these approaches are binary volume data sets.

Binary volume data only contains binary information (e.g., the points within the object have the value 1 and the background points have the value 0)

There are basically two techniques to extract a skeleton in discrete space:

**Topological Thinning** is an operation that iteratively peels the object layer by layer without destroying its topology, i.e. preserving its Euler number (for more details see [20]). The basic procedure of the algorithm is the deletion of simple points. A simple point is a point with neighborhood characteristics that ensure the topology preservation of the object after its deletion. Testing the simple point characteristic is a computationally expensive operation. Therefore, the Topological Thinning approach is time-consuming. On the other hand, this algorithm can be parallelized and the simple point calculation can be avoided in some cases. Thinning has the advantage that it has a mathematical foundation and the connectivity preservation can be proved. In section 4, this approach will be discussed in further detail.

**The Distance Transform Method** is based on the fact that the center of an object coincides with

points having maximal distance to the borders. A distance transform converts the binary data image into an image containing for every object point the minimum distance to the nearest background point. The local maximum of the transformed image represents a skeleton point. These methods have the disadvantage that they are not topology preserving. Post-processing must be applied to achieve topology preservation [41]. It is however less time consuming than the thinning operation.

**Camera motion** There are different approaches to control the camera in a virtual environment. Although a lot of research effort has been invested in this area, it is not yet fully explored [12][13][25].

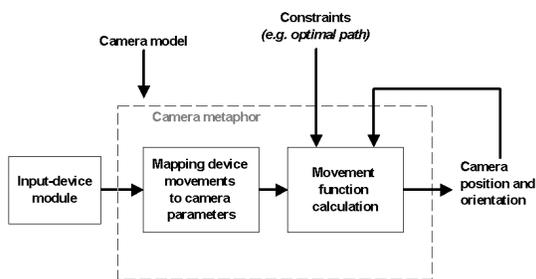


Figure 3: VirEn: Camera motion

**Planned navigation** is a technique already used in computer animation and robot path planning. Movements along the path are calculated offline. The user defines a certain number of keyframes where camera parameters are specified. Smooth camera movements between the keyframes are calculated automatically (e.g., using interpolation).

Another possibility is to automatically calculate the navigation path from a starting point to a target point previously defined by the user.

The drawback of planned navigation is the lack of interactivity, which can produce a tedious amount of work for the user to achieve the desired results .

**Manual navigation.** With manual navigation, the user has complete control over all parameters of the virtual camera without any constraints. The problem with this technique is the big number of parameters that the user has to control, which can get him easily disoriented.

**Guided navigation** is in between the two previous techniques. The user has control over the camera parameters but some constraints are added, such as keeping the position of the camera to the

optimal path. This implies losing some freedom in the interaction. Nevertheless, this approach avoids the lost-in-space feeling and interactivity is maintained [10]. Guided navigation is actually an extension of both planned and manual navigation. When the motion is too constrained, it is equivalent to planned navigation, and when there are no constraints, it will be like manual navigation.

The parameters that specify the position and orientation of the camera determine *the camera model* [8]. Some possible camera models include: PointLookAt [2] with the position of the camera and up vector; using Yaw/Roll/Pich angles for the orientation plus the camera position; or using quaternions and the camera position.

Viewpoint manipulation techniques are often called *camera metaphors*, which usually have a real world analogy (e.g., flying, walking, “scene-in-hand”, “eyeball-in-hand” [12][14]).

Camera motion in VirEn has been defined as shown in figure 3. It includes mapping of the device movements to the camera model parameters and can also include the calculation of movement functions. These functions can be used for example to provide smooth transitions or to apply constraints when guided navigation is used (e.g., keep the camera position in the optimal path, or forbid the camera to penetrate the walls using collision detection between the camera and the walls).

**Navigation structure.** A navigation structure is any data structure used in navigation in order to either accelerate rendering or to guide the camera motion. Navigation structures to accelerate rendering are broadly used in computer graphics applications like architectural walkthroughs and games [23][32][38], where polygonal rendering is used. These techniques could also be used in virtual endoscopy if surface rendering is used.

In the case of volume rendering, using the locality of the endoscopic view and the optimal path generation, subvolumes can be created to avoid having to process the whole volume in the rendering step.

In the case of branching organs like the bronchus having a tree structure can be useful to decide the navigation path and for accelerating the rendering process.

An endoscopic view just shows the local neighborhood of the camera position in detail. To provide the user with global information on his position within the entire object an *overview* is desirable. A sketchy representation of the entire object is sufficient for this purpose. The endoscopic view illustrates the focus area, while the overview illustrates the context area (see figure 9).

Another important point in virtual endoscopy is to provide the user with the possibility to realize *measurements*

(e.g., length, area, volume, distance) of interesting or pathological features.

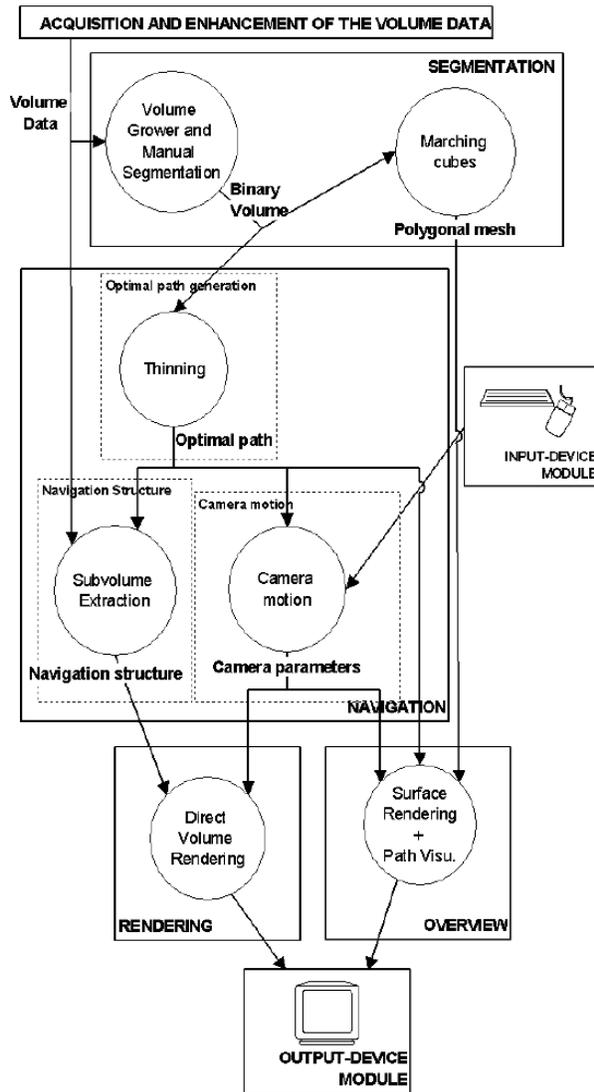


Figure 4: VirEn: Prototype structure

## 2.4 Current State

Currently, the authors are working on the development of the first prototype (shown in figure 4) based on the presented conceptual VirEn system. In the current state some of the described components have been developed using techniques described in the following. Segmentation is done using region growing [42] and manual segmentation. A marching cubes [22] algorithm is applied to the binary volume produced in the previous step to get the polygon mesh. The resolution of the volume is reduced in order to avoid large amounts of triangles. For the optimal path generation, a topological thinning algorithm has

been chosen (section 4). The camera motion follows the principles of guided navigation with free rotation movements. The position of the camera is restricted to the optimal path. Concerning the rendering process, a ray casting technique (shear warp factorization) with low resolution and sampling is used to achieve interactive update rates. In the overview image, the optimal path, the camera and the polygon mesh of the segmented surface is rendered (figure 9).

The result in the current state of the application can be seen in figure 9. In the following section an overview on some existing systems is given.

## 3 Related Work

A lot of work has been done in the last years related to virtual endoscopy. In this section some of the work developed by other authors is described.

Hong et al. [14] describes an endoscopy system which was used for colonoscopy. It is based on surface rendering. They use a modified region growing algorithm and marching cubes as segmentation techniques. Making use of the twisted shape of the colon, they employ potential visible sets and portals as navigation structures. As navigation metaphor, they use guided navigation with physically based camera control. A potential field is calculated using a distance transformation. This potential field is an attractive force to the target point while the walls act as repulsion force. A friction field is added in order to control the speed. The same research group improved the system in a later article [40] using direct volume rendering, which is accelerated by a distance transformation previously calculated for navigation. The algorithm is parallelized for acceleration. Yagel et al. [39] propose a system for the training of surgeons in virtual sinus surgery and for teaching in virtual endoscopy. Volume rendering is done using splatting [36] for high quality rendering and 3D texture mapping [5] in hardware for fast movements. They use special devices for interaction, like an endoscope device.

Geiger and Kikinis [11] present a virtual endoscopy system used in bronchoscopy for training and simulation of complicated cases. In order to segment the organ they use statistical segmentation or thresholding and also manual 2D contour editing. A Delaunay volume and surface reconstruction from the contours is done to be used in surface based rendering. For navigation a cylinder metaphor with the degrees of freedom of a real endoscope is used. The camera is constrained to turn around the axis of the cylinder and from 90 to -90 degrees in one axis perpendicular to the cylinder axis and also backward and forward movements are also allowed. Additionally the system detects penetration in which case a corrective motion is applied. Yet, this approach is not described in detail.

Two different virtual endoscopy systems are presented

by Robb [29]. In the article, he deals with both direct volume rendering and surface rendering. They also use two navigation methods: one is a planned navigation method used for direct volume rendering. For surface rendering, they use a fly-through method or a manual navigation method. In order to enhance the visualization stereoscopic vision and 3D devices are used.

Darabi and et al. [7] use volume rendering with ray casting and planned navigation. They suggest that stereographic projection gives a more realistic view since perspective projection fails because symmetric objects appear asymmetric near the boundaries.

In the proposal done by Shahidi et al. [31], segmentation is realized by thresholding, region growing and marching cubes. They utilize both manual navigation and planned navigation (keyframes), as surface rendering is used in this approach.

Paik et al. [28] describe a planned navigation approach using path planning. One of the main issues of this article is the automatic calculation of the medial axis using a thinning algorithm to automatically generate the navigation path. Segmentation is done using region growing, with multiple regions manually connected when a single one can not fill the entire structure.

Mori and Katada [26] propose the usage of surface rendering to implement a bronchoscopy system. A simple marching cubes algorithm is used for segmentation. A thinning algorithm generates a tree structure to obtain a skeleton. This skeleton is used to select the branch corresponding to the current viewpoint.

In [3] and [4] the authors describe a special type of perspective volume ray tracing which makes use of coherence between frames. A subvolumes system is utilized in order to deal with the huge data sets that the current imaging techniques are producing.

In the area of diagnosis there are completely different approaches. The organ volume is projected onto a plane where the inspection is done [35].

In this section several pieces of related work have been mentioned. In the next section the optimal path calculation of the VirEn prototype will be discussed.

## 4 Optimal path calculation

A crucial step in the prerequisites for intuitive navigation in VirEn is to obtain the optimal path. The thinning technique described by Ma and Sonka [24] has been extended. A good path is kept in the center of the hollow structure and connectivity is preserved. Therefore, the topological thinning approach was used despite of its computational effort. As the path generation can be done in a preprocessing step, execution time is less crucial. A thinning algorithm is an iterative process which deletes points with a special neighborhood characteristic, called *simple points* or *border points* [20]. It can be proven that deleting simple points preserves the Euler number and therefore the connectivity of the object.

In order to be geometry preserving (e.g., avoid that a tail disappears), a simple point is not deleted if it does not fit special geometric constraints. Geometric preservation is quite vaguely defined, and the different thinning algorithms differ in the way in which they define these geometric constraints.

The naive sequential thinning algorithm deletes one simple point at a time and it is quite straightforward to show that it preserves connectivity. A parallel thinning algorithm deletes a set of points in each iteration. This enables its simple parallelization and acceleration. Proving that a parallel thinning algorithm is topology preserving can be tricky. It has to be shown that when two adjacent points are deleted they can be deleted in a sequential way without affecting the simple point condition.

Various approaches on parallel thinning are described in specialized publications [9][17]. The thinning approach in VirEn has been based on the fully parallel 3D thinning algorithm proposed by Ma and Sonka in 1996 [24]. Fully parallel means that the algorithm is applied in parallel to all the object points of the image.

This algorithm, instead of calculating the simple point condition, checks the 26-neighborhood of the points with specific templates. The templates contain object points, background points and “do not care” points which can go either way.

The templates are obtained by rotations of the 4 so-called template cores (A, B, C, D) shown in figure 5. A template belongs to class A, B, C, or D if it has been generated by

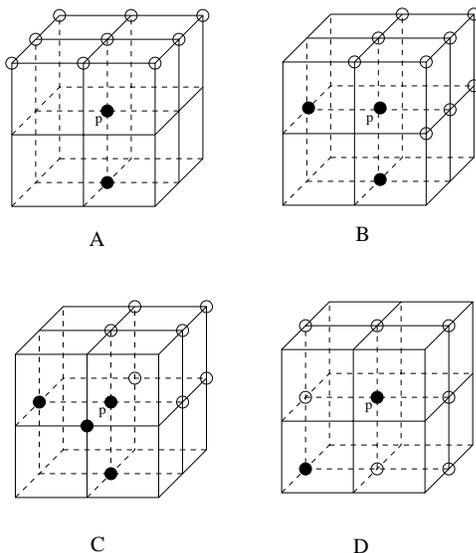


Figure 5: The template cores of the parallel thinning algorithm: the unmarked point is a “do not care point”, white points are background points and black points are object points. For the D core template, p must be a simple point.

the template core A, B, C or D respectively. Checking the 26-neighborhood is not sufficient to preserve connectivity, and several templates must check extra points in an extended neighborhood. In figure 6, the six templates of class A (figure 5) and the extra tests needed in those templates are illustrated.

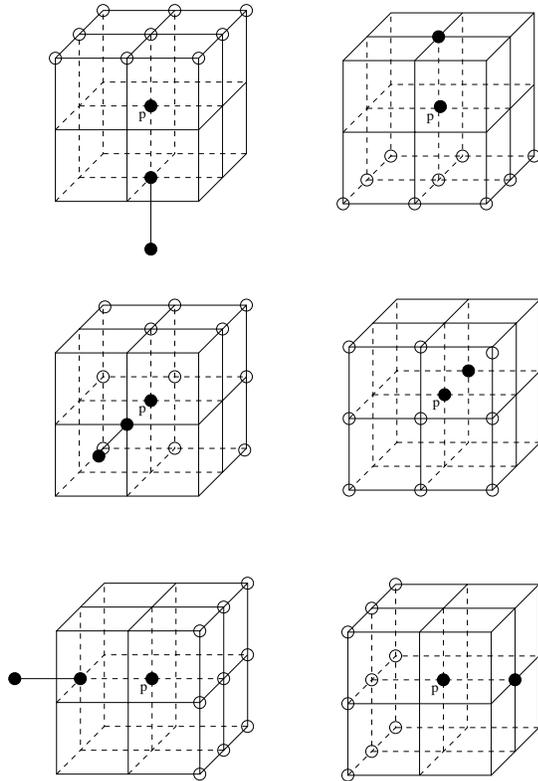


Figure 6: The six templates for deleting border points obtained from the core template A.

In our implementation, the 26-neighborhood of a point and the templates are represented by masks. The masks are a chain of 27 binary values (26-neighborhood and the point) where values are: 01 for object points, 10 for background points and 11 for “do not care” points. A binary operation (*AND*) and a comparison are enough to find out whether a point fulfills a template condition or not. In various cases tests to an extended neighborhood have to be done. This fact increases computation time. Nevertheless, only the templates generated from the template core D (figure 5) require the time-consuming simple point test.

This algorithm presents some problems. One problem arises when a point that fulfills the conditions of the class D templates is deleted in parallel with some of its adjacent points. It has to be proven that they can be deleted in a sequential way without breaking the class D template conditions and so for the simple point test. This cannot be proven without restricting the simple point condition to a subgroup of all the possible cases. The effect of this restriction can be seen in figure 8a compared with

figure 8b. In figure 8b, the templates of class D are treated in a separate iteration. Although this partly loosens upon parallelism, the results are better (see figure 8). Therefore, the authors decided to use this last approach.

Another problem of the algorithm is that the extended neighborhood tests are not symmetric and therefore the algorithm is orientation dependent. The algorithm does not treat borders of one side of the object and borders of the opposite side the same way. This problem is illustrated in figure 8b. Although the object represented is symmetric the thinning result is not. This orientation dependency can be softened by permuting the orientation of the extended templates in each iteration of the algorithm (see figure 8c). It is straightforward to see that this change does not affect the mathematical verification of the connectivity preservation presented by Ma and Sonka [24].

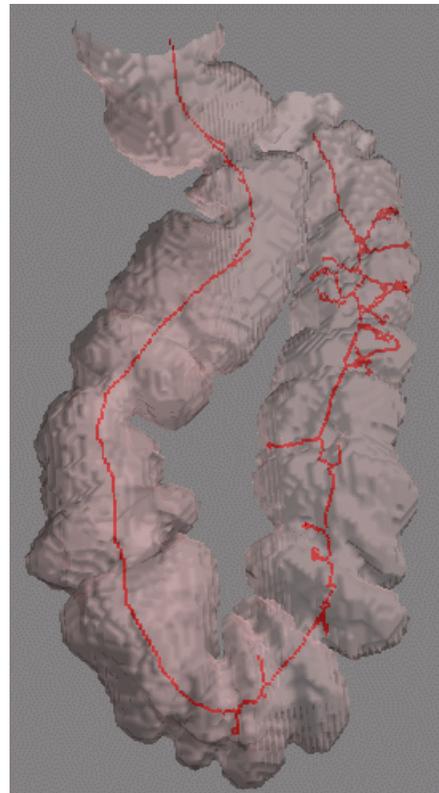


Figure 7: Skeleton obtained by the topological thinning together with a transparent surface of the binary segmented data.

Some results using a real data set (Intestine piece data set 198x115x300) are shown in figure 7. Since the real objects are not regular the skeletons usually are noisy and have some little branches because of the geometry preservation. Therefore the skeleton must be filtered to remove these branches (see figure 7).

## 5 Implementation

The VirEn project is part of the VisMed project (<http://www.vismed.at>) which is developed on the PC platform running Windows NT. The application is being developed using the object oriented paradigm and C++ as a programming language.

## 6 Conclusions and Future Work

The main point of this paper is the presentation of a framework for virtual endoscopy developed by the authors: VirEn. The most important and critical components for a virtual endoscopy system have been discussed. A basic prototype based on VirEn has been implemented. Special effort has been invested in the development of the navigation and optimal path calculation features. An extension to a thinning algorithm based on the work by Ma and Sonka [24] has been presented. Resulting images proved the usability of this approach.

The extension of the basic endoscopy framework will be addressed as future work. Research and implementation effort will concentrate on direct volume rendering and navigation structures. The rendering speed / quality trade-off will have to be optimized. Additional work will be invested on the development of an enhanced camera motion model, which will complete the requirements for a virtual endoscopy system capable of being used in daily clinical practice.

## Acknowledgements

The work presented in this publication has been funded by the VisMed project. VisMed is supported by *Tiani Medigraph*, Vienna, <http://www.tiani.com>, and the *Forschungsförderungsfonds für die gewerbliche Wirtschaft*, Austria, <http://www.telekom.at/fff/>. Please refer to <http://www.vismed.at> for further information on this project.

## References

- [1] L. Bergman, H. Fuchs, E. Grant, and S. Spach. Image rendering by adaptive refinement. In David C. Evans and Rusell J. Athay, editors, *SIGGRAPH 86 Conference Proceedings*, Annual Conference Series, pages 29–37. ACM SIGGRAPH, Addison Wesley, August 1986.
- [2] J. Blinn. Where am I? what am I looking at? *IEEE Computer Graphics and Applications*, 8(4):76–81, July 1988.
- [3] M. L. Brady, K. Jung, H. T. Nguyen, and T. Nguyen. Two-phase perspective ray casting for interactive volume navigation. In *IEEE Visualization '97*, pages 183–189, October 1997.
- [4] M. L. Brady, K. K. Jung, H. T. Nguyen, and T. PQ. Nguyen. Interactive volume navigation. *IEEE Transactions on Visualization and Computer Graphics*, 4(3):243–255, July – September 1998.
- [5] B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In Arie Kaufman and Wolfgang Krueger, editors, *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, October 1994.
- [6] J. Danskin and P. Hanrahan. Fast algorithms for volume ray tracing. In *Proceedings of the Workshop on Volume Visualization*, pages 91–98, New York, October 19–20 1992. ACM Press.
- [7] K. Darabi, K. D. M. Resch, J. Weinert, and U. Jendrysiak. Real and simulated endoscopy of neurosurgical approaches in an anatomical model. *Lecture Notes in Computer Science*, 1205:323–326, 1997.
- [8] S. M. Drucker, T. A. Galyean, and D. Zeltzer. CINEMA: A system for procedural camera movements. In Marc Levoy and Edwin E. Catmull, editors, *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pages 67–70, Cambridge, MA, March–April 1992. ACM Press.
- [9] N. Gagvani. Skeletons and volume thinning in visualization. Master's thesis, New Brunswick, Rutgers , The State University of New Jersey, 1997.
- [10] T. A. Galyean. Guided navigation of virtual environments. In ACM-Press, editor, *1995 Symposium on Interactive 3D Graphics*, pages 103–104. ACM SIGGRAPH, April 1995.
- [11] B. Geiger and R. Kikinis. Simulation of endoscopy. In Nicholas Ayache, editor, *Computer Vision, Virtual Reality and Robotics in Medicine*, Lecture Notes in Computer Science, pages 276–282. Springer-Verlag, April 1995.
- [12] C. Hand. Survey of 3D interaction techniques. *Computer Graphics Forum*, 16(5):269–281, December 1997.
- [13] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. A survey of design issues in spatial input. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, Two Hands and Three Dimensions, pages 213–222, 1994.
- [14] L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual voyage: Interactive navigation in the

- human colon. In *SIGGRAPH 97 Conference Proceedings*, Annual Conference series, pages 27–34. ACM SIGGRAPH, Addison Wesley, August 1997.
- [15] R. B. Jeffrey. Three-dimensional imaging and virtual reality application of spiral ct. In G.P. Krestin and G.M. Glazer, editors, *Advances in CT IV*, volume 4, conference book 18, pages 149–153. Springer, March 1998.
- [16] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. In *First International Conference on Computer Vision, (London, England, June 8–11, 1987)*, pages 259–268. IEEE Computer Society Press, 1987.
- [17] T.Y. Kong. One problem of determining whether a parallel reduction operator for n-dimensional binary images always preserves topology. In *Proceeding in SPIE Conference on Vision Geometry*, pages 69–77, 1993.
- [18] P. Lacroute and M. Levoy. Fast volume rendering using a shear-warp factorization of the viewing transformation. In *SIGGRAPH 94 Conference Proceedings*, Annual Conference Series, pages 451–458. ACM SIGGRAPH, Addison Wesley, 1994.
- [19] M. Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(3):29–37, February 1987.
- [20] G. Lohmann. *Volumetric Image Analysis*. Chichester Wiley, 1998.
- [21] B. Lorensen, R. Kikinis, J. Flynn, A. Kaufman, and S. Napel. Surface rendering versus volume rendering in medical imaging: Techniques and applications (panel). In Roni Yagel and Gregory M. Nielson, editors, *Proceedings of the Conference on Visualization*, pages 439–440, October 27–November 1 1996.
- [22] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In Maureen C. Stone, editor, *SIGGRAPH 87 Conference Proceedings*, Annual Conference Series, pages 163–169. ACM SIGGRAPH, Addison Wesley, July 1987.
- [23] D. Luebke and C. Georges. Portals and mirrors: Simple, fast evaluation of potentially visible sets. In Pat Hanrahan and Jim Winget, editors, *1995 Symposium on Interactive 3D Graphics*, pages 105–106. ACM SIGGRAPH, April 1995.
- [24] C. Min Ma and M. Sonka. A fully parallel 3d thinning algorithm and its application. In *Computer Vision Graphics and Imaging Processing: image Understanding, Conference Proceedings*, volume 64, pages 420–433, 1996.
- [25] J. D. Mackinlay, S. K. Card, and G. G. Robertson. Rapid controlled movement through a virtual 3D workspace. In Forest Baskett, editor, *SIGGRAPH 90 Conference Proceedings*, Annual Conference Series, pages 171–176. ACM SIGGRAPH, Addison Wesley, August 1990.
- [26] K. Mori, J. Hasegawa, J. Toriwaki, H. Anno, and K. Katada. A fast rendering method using the tree structure of objects in virtualized bronchus endoscope system. *Lecture Notes in Computer Science*, 1131:33–42, 1996.
- [27] E. N. Mortensen and W. A. Barrett. Intelligent scissors for image composition. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 191–198. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.
- [28] D. S. Paik, C. F. Beaulieu, R. B. Jeffrey, G. D., Rubin, and S. Napel. Automated path planning for virtual endoscopy. *Medical Physics*, 25(5):629–637, May 1998.
- [29] R. A. Robb. Virtual (computed) endoscopy: Development and evaluation using the visible human datasets. Mayo Foundation/Clinic, October 1996.
- [30] P. Schröder and G. Stoll. Data parallel volume rendering as line drawing. In *Proceedings of the Workshop on Volume Visualization*, pages 25–32, New York, October 19–20 1992. Acm Press.
- [31] R. Shahidi, V. Argiro, S. Napel, L. Gray, H. P. McAdams, G.D. Rubin, C F Beaulieu, R.B. Jeffrey, and A. Johnson. Assessment of several virtual endoscopy techniques using computed tomography and perspective volume rendering. *Lecture Notes in Computer Science*, 1131:521–526, 1996.
- [32] S. J. Teller and C. H. Séquin. Visibility preprocessing for interactive walkthroughs. In Thomas W. Sederberg, editor, *SIGGRAPH 91 Conference Proceedings*, Annual Conference Series, pages 61–69. ACM SIGGRAPH, Addison Wesley, July 1991.
- [33] H. K. Tuy and L. T. Tuy. Direct 2-D display of 3-D objects. *IEEE Computer Graphics and Applications*, 4(10):29–33, October 1984.
- [34] C. Upson and M. Keeler. V-BUFFER: Visible volume rendering. In John Dill, editor, *SIGGRAPH 88 Conference Proceedings*, Annual Conference Series, pages 59–64. ACM SIGGRAPH, Addison Wesley, August 1988.
- [35] G. Wang, M. W. Vannier, E. McFarland, and J. P. Heiken. GI tract unraveling in volumetric CT. *Lecture Notes in Computer Science*, 1131:3–12, 1996.

- [36] L. Westover. Footprint evaluation for volume rendering. In Forest Baskett, editor, *SIGGRAPH 90 Conference Proceedings*, Annual Conference Series, pages 367–376. ACM SIGGRAPH, Addison Wesley, August 1990.
- [37] R. Yagel. Towards real time volume rendering. In *Proceedings of GRAPHICON'96*, pages 230–241, July 1996.
- [38] R. Yagel and W. Ray. Visibility computation for efficient walkthrough of complex environments. *PRES-ENCE*, 5(1):1–16, 1996.
- [39] R. Yagel, D. Stredney, G. J. Wiet, P. Schmalbrock, L. Rosenberg, D. J. Sessanna, and Y. Kurzion. Building a virtual environment for endoscopic sinus surgery simulation. *Computers and Graphics*, 20(6):813–823, December 1996.
- [40] S. You, L. Hong, M. Wan, K. Junyaprasert, A. Kaufman, S. Muraki, Y. Zhou, M. Wax, and Z. Liang. Interactive volume rendering for virtual colonoscopy. In Roni Yagel and Hans Hagen, editors, *IEEE Visualization '97*, pages 433–346. IEEE, November 1997.
- [41] Y. Zhou, A. Kaufman, and A. W. Toga. Three-dimensional skeleton and centerline generation based on an approximate maximum distance field. *Visual Computer*, 14:303–314, 1998.
- [42] S. W. Zucker. Region growing: Childhood and adolescence. *Computer Graphics and Image Processing*, 5(3):382–399, September 1976.

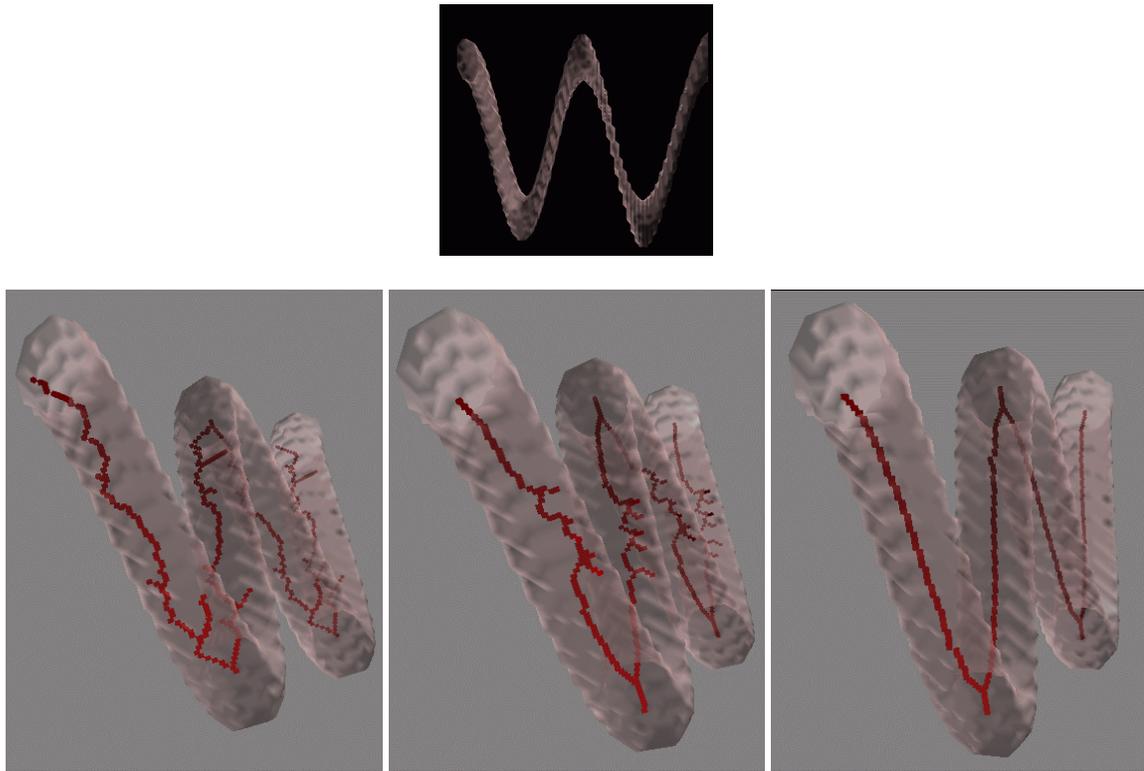


Figure 8: Comparison between the different modifications applied to the thinning algorithm proposed by Ma and Sonka [24]: in the top the artificial data set used for the comparison; **a) left** Original algorithm [24] result. **b) middle** Not fully parallel algorithm result. **c) right** The same algorithm as *b* plus avoiding orientation dependencies.

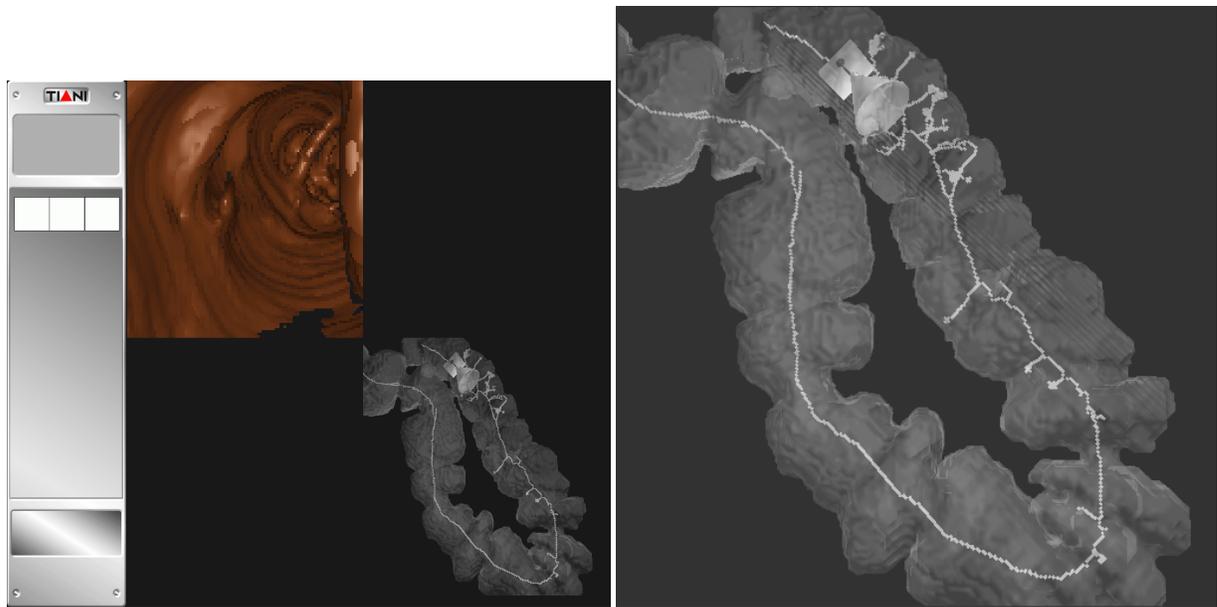


Figure 9: VirEn Prototype: **a) left** Application snapshot. **b) right** Enlargement of the overview. There can be seen the surface of the object, the skeleton and the camera. The camera is represented by its frustum, image plane and up vector.