VIVENDI - A Virtual Ventricle Endoscopy System for Virtual Medicine*

Dirk Bartz[1] and Martin Skalej[2]

[1] WSI/GRIS, University of Tübingen,
Auf der Morgenstelle 10/C9,
D72076 Tübingen, Germany
Email: bartz@gris.uni-tuebingen.de
[2] Department of Neuroradiology, University Hospital Tübingen
Hoppe-Seyler-Str. 3
D72076 Tübingen, Germany

Abstract. Virtual Medicine is an emerging and challenging field in Computer Graphics. Numerous visualization methods are used to model and render data of different modalities.
In this paper, we present a new endoscopy system for virtual medicine. The main purpose of this system is to provide support for the planning of complicated endoscopic interventions inside of the ventricular system of the human brain. Although, our current focus is on ventricle endoscopy, this system is applicable to other areas as well.
In order to achieve interactive framerates on workstations with medium graphics performance, we apply an efficient implementation of a basic algorithm for general visibility queries.
Keywords: Virtual Medicine, Virtual Environments, Surgical Assist Systems.

1 Introduction

Minimally-invasive neurosurgical procedures are of increasing importance in neurosurgery. In comparison to commonly used surgical techniques, less brain tissue is damaged. Furthermore, minimally-invasive procedures have less deleterious effects on the patient. On the other hand, these procedures lack fast access in case of serious complications, such as strong bleeding. Therefore, careful planning and realization of this procedure is essential, in order to avoid such complications. This problem aggravates, because handling and control of these endoscopes is very difficult, mainly due to limited flexibility of and limited field of view through the endoscope (which provides only a very limited orientation), and the sensitive nature of the brain tissue.

To optimize the success of these interventions, an improved planning and training environment is required. Consequently, we developed a virtual ventricle endoscopy system in cooperation with the Department of Neuroradiology of the university hospital at Tübingen.

* *To appear in the Symposium on Visualization 1999 proceedings*

Traditional planning of endoscopic interventions in neurosurgery is based on the thorough examination of slice images of a MRI scan. We add two additional stages to this planning pipeline; a pre-processing stage and the virtual endoscopy of the ventricular system (Figure 1).

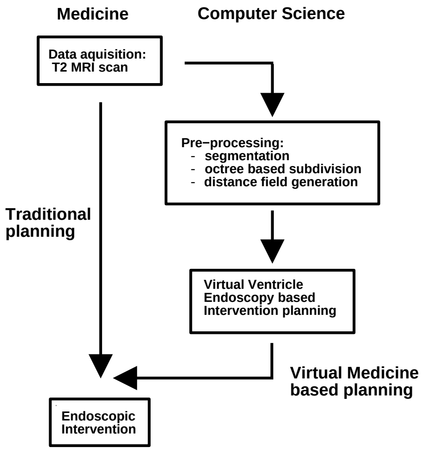

Medicine
Computer Science
Data aquisition:
T2 MRI scan
Pre−processing:
- segmentation
- octree based subdivision
- distance field generation
Traditional planning
Virtual Ventricle Endoscopy based Intervention planning
Virtual Medicine based planning
Endoscopic Intervention

Fig. 1. Flow of virtual endoscopy-based intervention planning.

We will discuss these two stages in this paper, which is organized as follows: We start with a brief survey of methods for virtual medicine. In Section 3, we outline the basic medical problems that indicate the need for ventricle endoscopy. In Section 4, we present our system for planning endoscopic interventions inside the ventricular system. Finally, in Section 5, we state a conclusion and give perspectives to future work.

2 Related Work

There has been quite some work in the field of virtual medicine. In this Section, we present a brief discussion of some of the related work. Among the first is [17], where methods of Computer-Aided Geometric Design (CAGD) were applied for planning of cranial surgery.

In [13, 10, 9], Finite Element Methods were used to construct models of tissue of the human body. These models were used to simulate deformations of the tissue to predict the outcome of plastic or craniofacial surgery.
Computer-based anatomical atlases were introduced by Höhne et al. [5]. Based on CT and MRI volume data, methods of artificial intelligence and volume rendering were combined.

Recently, Serra et al. introduced a framework for stereo-tactic frame surgery [16], using a set of computer-based tools and a 3D-output device, similar to the virtual table paradigm.

One of the major recent research topics in virtual medicine is virtual endoscopy. Frequently, the proposed methods generate off-line animations of virtual cameras simulating an endoscopic session through various hollow organs. In 1994, Geiger and Kikinis proposed using Finite Element Methods to specify a path of the camera [3]. Similar approaches of automatic generation of the camera path were investigated by Vining et al. [18], Lorensen et al. [11], and Hong [6]. In contrast, Rubin et al. manually specify a key-frame interpolated camera path [14].
Hong et al. proposed a new navigation method, implementing the guided-navigation paradigms [7]. By combining distance fields and kinematic rules, an intuitive scheme for navigating inside the human colon was developed. Furthermore, a customized visibility algorithm was proposed in order to reduce the number of surface polygons of the inner surface of the colon to a feasible size. While this system provided a fast and intuitive handling of the virtual endoscope, it required high-end SGI InfiniteReality graphics for interactive framerates.

3 Ventricle Endoscopy

In minimally-invasive neurosurgery of the brain, existing cavities can be used as a preformed path for movements of the endoscope, without destruction of brain tissue. Our focus is on the ventricular system of the human brain, in which the brain liquor (cerebrospinal fluid or CSF) is produced and resorbed (Figure 2 (a)). To access the ventricles, a hole is drilled through the skull and a tube is placed through this hole into the ventricular system. Thereafter, the endoscope is introduced through the tube, which is used as a stable guide for the endoscope.

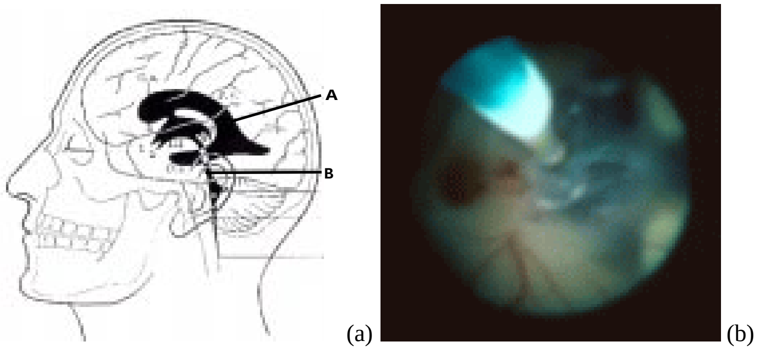


Fig. 2. Ventricular system of the human head: (a) A ventricles, B aqueduct; (b) view through endoscope.

Because of the water-like optical property of the CSF - which fills the ventricular system, viewing of the surrounding tissue is possible. Movement of the endoscope - guided by video-control via the small field of view of the endoscope - is limited by the tube and the surrounding tissue (Figure 2 (b)). Micro-instruments, introduced through an additional canal inside the endoscope, can then be used to perform the actual minimally-invasive procedure, e.g. removing accessible mass lesions.

Due to respiration and other metabolistic activity, the CSF flows through the cavities inside the human brain. In some cases, the drain inside the ventricles - the aqueduct (Figure 2 (a), B) - is blocked by occlusion or stenosis. This causes a serious disturbance of the natural flow of the CSF, which frequently leads to a dangerous increase of pressure inside the skull and can damage the brain severely. The clinical picture of this hydrocephalus is one of the major indications for a minimally-invasive intervention in the ventricular system, where a bypass is realized by perforating the floor of the third ventricle. However, the limited view and orientation through-out the intervention increases the necessary time of the intervention and consequently, the inherent risks of serious complications. To overcome these drawbacks, we propose the use of a virtual endoscopy system to improve the planning of and orientation during this procedure.

4 VIVENDI - Virtual Ventricle Endoscopy

In this Section, we discuss the elements of the VIVENDI system for virtual ventricle endoscopy. In some parts, VIVENDI follows the VICON system for interactive colonoscopy [7]. However, due to the different anatomical topology, we use a different subdivision and visibility scheme. Therefore, the only common method is the guided-navigation system, which is already discussed in detail in [7].

4.1 System Architecture

The endoscopy system itself consist of two stages: pre-process and interactive virtual endoscopy. The pre-processing stage is responsible for the generation of numerous auxiliary data, which is later used during the interactive virtual endoscopy. It is organized in three major steps, which are outlined in Figure 3. In the first step, we segment the voxels classified as inside of the ventricles using a 3D region growing algorithm starting from an interactively specified seed point. Subsequently, the default path of our virtual camera is generated using Dijkstra's minimum path algorithm [7], using the seed point and an additionally specified target point as start and end point of the camera path. Thereafter, we extract the isosurfaces of the ventricular system, using an octree-based parallel implementation of the Marching Cubes algorithm [1]. The size of the leaf blocks of the octree depends on a specified granularity value of volume cells which intersect with the isosurface[1]. Based on this octree representation, a subdivision of the extracted isosurface is generated, where the isosurface geometry associated with an octree leaf block is considered as one subdivision entity. Finally, three distance fields are

[1] We call these cells relevant cells and the respective granularity value *relevant cell load* or RCL.

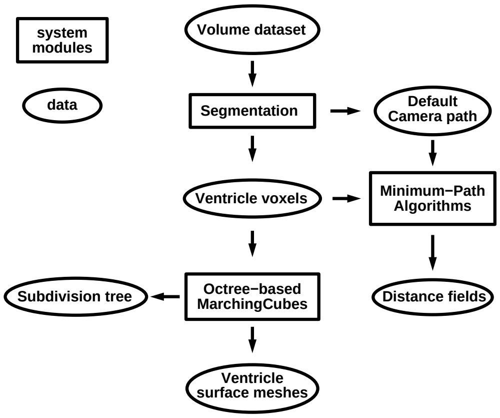


Fig. 3. Pre-process flow.

computed, implementing a collision avoidance scheme and the motion of the virtual camera for guided-navigation [7] towards a target point.

For interactive colonoscopy, complete pre-processing time took up to ten hours. Most of the time was spent on generating the default camera path (or skeleton of the colon) and on the generation of the subdivision along this skeleton. However, this time can be greatly reduced by using improved data structures and algorithms. Replacing FIFO-queue based priority queues of the first pre-processing step by Fibonacci heaps [12], we could reduce the algorithmic time complexity, hence reducing the time consumption significantly.
For the previous skeleton-based subdivision step, a back-tracking algorithm was used, in order to optimize the size of the respective subdivision cells. However, this was a computational expensive approach, which frequently lead to a processing time of several hours. Due to the shape of the ventricular system, a tube-like subdivision is not available. Instead, we used a generalized octree-based subdivision scheme where computational costs are only a fraction of the skeleton-based subdivision. In total, we reduced the pre-processing time down to approximately 15 minutes.

The interactive endoscopy stage of the VIVENDI system is built from an user-interface and system core (Figure 4). The centerpiece of the latter is the VIVENDI render system (module VIVENDI Rendering), which is responsible for the OpenGL rendering of the geometry of the organ (Fig. 5). In order to reduce its complexity, a visibility culling algorithm is applied (module Visibility, see Section 4.2 for details). User-interaction (for navigation and measurement) via the rendering area of the VIVENDI rendering is bypassed to the Main Control Panel. Position and orientation of the virtual camera are provided by the guided-navigation system (module Camera, see [7] for details). If the user initiates direct volume rendering (module Volume Renderer), the generated images are overlaid with the polygonal rendering of VIVENDI Rendering.

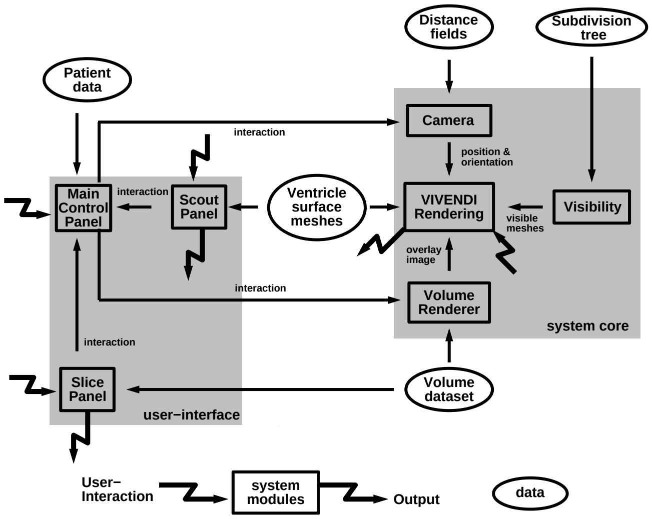


Fig. 4. VIVENDI control flow.

On the left hand side of Figure 4, the user-interface (see Fig. 5) is organized around the Main Control Panel, which provides control over camera navigation, volume rendering, animation generation, and other general parameters of the system. The Main Control Panel communicates with the Slice Panel, which provides the traditional three orthogonal slices of the volume dataset in different resolutions. The Scout Panel provides a fully rotatable 3D overview of the surface of the ventricular system. Furthermore, it administrates the position markers and controls the multiple camera paths (Section 4.3). If a position marker is used as a jump mark, it provides the virtual camera with the new position and orientation via the Main Control Panel.

4.2 Visibility

Research on algorithms for visibility queries is one of the most active areas in Computer Graphics. Consequently, several algorithms for occlusion culling have been proposed. Most notable are the hierarchical Zbuffer (HZB) approach [4], hierarchical occlusion maps (HOM) [19], and the aggregate cull rectangle algorithm (ACR) [7]. Considering the structure of the problem and necessary interactivity, none of these algorithms are suited for virtual ventricle endoscopy. The HZB-algorithm lacks efficiency for interactive rendering of medium sized datasets (up to 1M triangles) on current graphics computers. In contrast, the HOM-algorithm requires a dedicated scene subdivision into separated objects in order to get a good occluder pre-selection, which is necessary for high performance. Finally, the ACR-algorithm provides a good interactive performance, based on a tube-like topology — like the human colon. However, this topology

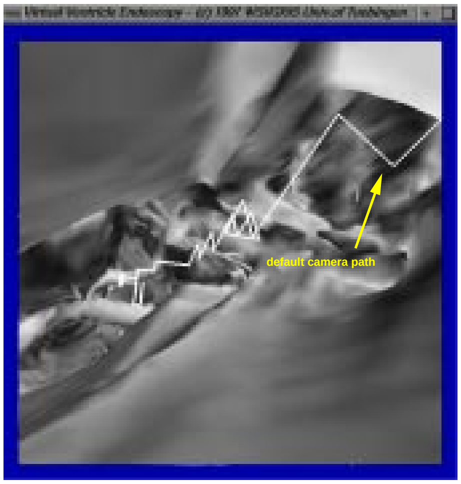

Virtual Ventricle Endoscopy - (c) 1997 WSI/GRIS Univ. of Tuebingen
default camera path

Fig. 5. Endoscopic view.

is not available in the ventricular system. For these reasons, we chose view-frustum culling [2] for our application.

After subdividing the ventricle dataset into octree blocks in a pre-process step, each octree block of the subdivision tree above a given granularity of relevant cells (RCL) is tested for intersection with the view-frustum. If the octree block is completely within the view-frustum, all its child blocks are within the view-frustum as well, hence they are potentially visible. If the octree block is only partially inside the view-frustum, we proceed with its child blocks. If this block has no further children — it has reached the specified polygon load —, all its polygons are considered potentially visible. Finally, if the octree block is completely outside the view-frustum, all its child blocks and associated polygons are not visible and therefore, not rendered.

We exploit the *OpenGL selection mode* to implement our culling algorithm [8]. The whole view-frustum is used as the sensitive area for selection. We render the named (with an ID) octree blocks in selection mode, which generates no contribution to the framebuffer. If any geometry of the octree blocks intersects with the view-frustum, the selection mode returns a *hit report*, including the IDs associated with these blocks and their minimum z-values. If a block completely contains the view-frustum, this simple test would fail. Therefore, it is necessary to explicitly check before-hand for this case.

Recently, Hewlett-Packard introduced their fx-series of graphics systems. A speciality of this system is the support for occlusion culling with the *HP Occlusion Culling Flag* [15]. We rendered the depth-sorted (according to their minimum z-value) potentially visible octree blocks using this functionallity on a HP B180/fx4 graphics workstation. In a special *occlusion mode*, the HP flag detects if a rendered geometry would change the content of the z-buffer of the framebuffer. By checking this flag, we can determine if the octree block is occluded.

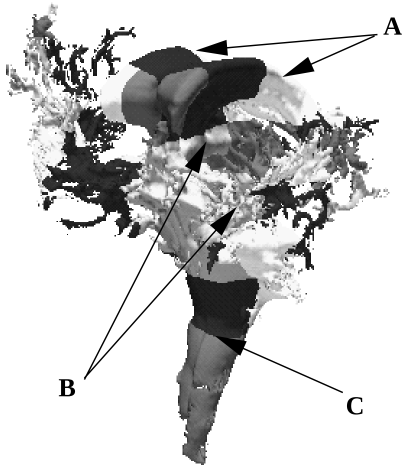


Fig. 6. Octree Subdivision of Ventricular system - different blocks are marked with different grey-levels. Note that the segmentation does not only select voxels of the ventricular system, but all connected CSF-filled cavities. A: Lateral Ventricles, B: 3rd and 4th Ventricles, C: Central Canal.

Table 1 shows the trade-off between visibility test overhead and rendering performance. The first subdivision into 120 blocks achieves the best framerate. However, the framerate of the second subdivision is only 0.4 frames/second (fps) lower, although the number of rendered polygons is 15% larger. Furthermore, Table 1 show the results of view-frustum culling with and without occlusion culling on a HP B-class fx4 workstation using a 180 MHz PA-7300 LC CPU and the HP Occlusion Culling flag. Overall, a culling performance of up to 93% of the model and framerates up to 19.4 fps on the HP were achieved. However, in some areas with low occlusion, occlusion culling does not increase the framerate. In fact, it might reduce the framerate if the occlusion culling overhead exceeds the benefits of culling polygons of the model. Although we only presented results for one dataset, other datasets showed a similar performance.

4.3 Multiple Camera Paths

Segmentation and navigation depend heavily on the start point of the camera path (the seed point of the segmentation) and the camera path itself respectively. However, not all areas of interest are connected by voxels classified as inside. This is due to partial volume effects, lack of resolution or obstruction of narrow areas. Furthermore, some examinations require different camera paths in order to reach different locations within the ventricular system. For these cases, VIVENDI supports multiple camera paths, which combines the models of different areas of interest of one volume dataset into a joined model. Each single model is pre-processed individually - providing its own

RCL	total #blocks	% blocks visible	#polygons visible	cull rate [%]	framerate [fps]
View-frustum culling only					
1000	120	52	185,834	52.4	7.6
2000	59	27	220,414	43.7	7.14
4000	28	16	248,618	36.5	7.12
View-frustum and occlusion culling					
1000	120	45	27,121	93.1	19.4
2000	59	13	50,054	87.2	17.8
4000	28	13	83,941	78.6	15.5

Table 1. Average performance of view-frustum culling of a ventricular system with more than 391K polygons.

camera path (and reconstructed isosurface if necessary) - and finally combined into the joined model with the respective number of camera paths. Alternatively, this functionality can be used to generate multiple camera paths in a single model[2].

For the actual virtual endoscopic examination, all individual models of the joined model which are not the current one, are considered not visible. To change to one of the other camera paths, automatic pre-defined blue model markers at the ends of the camera paths on the scout panel can be selected (Figure 7). In the course of changing to another camera path, the distance fields are changed too. This is necessary to reduce the look-up time of the binary tree on the compressed representation of the distance fields.

5 Conclusion and Future Work

In this paper, we presented a virtual endoscopy system, applied to virtual endoscopy of the ventricular system of the human brain. Starting from a virtual colonoscopy system, we replaced a visibility algorithm for tube-like objects by a basic view-frustum culling method, due to the non-tube topology of the ventricular system. Furthermore, we improved and adapted the pre-processing steps of our application. Pre-processing time of several hours was reduced to a few minutes. Finally, we introduced multiple camera paths - combining and handling of several camera paths in one model - and a new user-interface to adapt to the needs of our partners at the hospital.

Our system is suitable to many applications for virtual endoscopy, although the presented application is virtual ventricle endoscopy to support planning of endoscopic interventions. Therefore, one of the future focuses will be on further exploration of other applications of virtual endoscopy, especially inside the human head.

Early feedback from the Department of Neurosurgery suggests a high usability and acceptance for this clinical application. Nevertheless, further clinical studies are on the

[2] In this case, the single model is not simply copied, but referenced for each additional camera path.

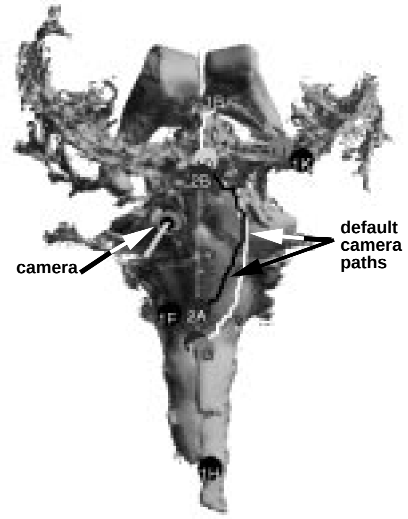


Fig. 7. Snapshot from the Scout Panel: Two different camera paths are combined into one model. Blue (dark) markers are pre-defined model markers; red, green, and yellow (different shades of grey) markers are user-defined markers.

agenda of future work.
For our measurements, we used a HP B180/fx4 workstation. However, measurements of smaller ventricular system models suggest that computer systems with less rendering performance and without hardware support for occlusion culling, i.e., a SGI O_2 or a PC graphics card, have sufficient graphics performance if powerful standard OpenGL-based occlusion culling algorithms are used[8]. Consequently, a major research focus will be on occlusion culling methods.

Acknowledgments

This work has been supported by the MedWis program of the German Federal Ministry for Education, Science, Research and Technology, and by hardware of the Hewlett-Packard Workstation Systems Lab., Ft. Collins, CO. Early implementations were derived from a prototype of the VICON system of the Center of Visual Computing at Stony Brook.

Datasets were provided by the Department of Neuroradiology of the University Hospital Tübingen. Special thanks to Dorothea Welte, Barbara Kortmann, Rupert Kolb, and Mechthild Uesbeck of the Department of Neuroradiology, Frank Duffner of the Department of Neurosurgery, and Hendrik-Jan van Veen of the Max-Planck-Instiute for Biological Cybernetics. Last but not least, we thank Michael Doggett and Michael Meißner for proof-reading.

References

1. D. Bartz, R. Grosso, T. Ertl, and W. Straßer. Parallel Construction and Isosurface Extraction of Recursive Tree Structures. In *Proc. of WSCG*, volume III, 1998.
2. B. Garlick, D. Baum, and J. Winget. Interactive Viewing of Large Geometric Databases Using Multiprocessor Graphics Workstations. In *ACM SIGGRAPH'90 course notes: Parallel Algorithms and Architectures for 3D Image Generation*, 1990.
3. B. Geiger and R. Kikinis. Simulation of Endoscopy. In *AAAI Spring Symposium Series: Application of Computer Vision in Medical Image Processing*, pages 138–140, 1994.
4. N. Greene, M. Kass, and G. Miller. Hierarchical Z-Buffer Visibility. In *Proc. of ACM SIGGRAPH*, pages 231–238, 1993.
5. K. Höhne, M. Bomans, M. Riemer, R. Schubert, and U. Tiede. A 3D Anatomical Atlas Based on a Volume Model. *IEEE Computer Graphics Applications*, 12:72–78, 1992.
6. L. Hong, A. Kaufman, Y. Wei, A. Viswambharan, M. Wax, and Z. Liang. 3D Virtual Colonoscopy. In *IEEE Symposium on Biomedical Visualization*, pages 26–32, 1995.
7. L. Hong, S. Muraki, A. Kaufman, D. Bartz, and T. He. Virtual Voyage: Interactive Navigation in the Human Colon. In *Proc. of ACM SIGGRAPH*, pages 27–34, 1997.
8. T. Hüttner, M. Meißner, and D. Bartz. OpenGL-assisted Visibility Queries of Large Polygonal Models. Technical Report WSI-98-6, ISSN 0946-3852, Dept. of Computer Science (WSI), University of Tübingen, 1998.
9. E. Keeve, S. Girod, P. Pfeifle, and B. Girod. Anatomy-based Facial Tissue Modelling Using the Finite Element Method. In *Proc. of IEEE Visualization*, pages 21–28, 1996.
10. R. Koch, M. Gross, F. Carls, D. Büren, G. Fankhauser, and Y. Parish. Simulating Facial Surgery Using Finite Element Methods. In *Proc. of ACM SIGGRAPH*, pages 421–428, 1996.
11. W. Lorensen, F. Jolesz, and R. Kikinis. The Exploration of Cross-Sectional Data with a Virtual Endoscope. In R. Satava and K. Morgan, editors, *Interactive Technology and New Medical Paradigms for Health Care*, pages 221–230. 1995.
12. K. Mehlhorn, S. Näher, and C. Uhrig. The LEDA Platform for Combinatorial and Geometric Computing. In *24th International Colloquium on Automata, Languages, and Programming (ICALP-97)*, volume LNCS 1256, pages 7–16, 1997.
13. S. Pieper. *CAPS: Computer Aided Plastic Surgery*. PhD thesis, MIT, 1992.
14. G. Rubin, C. Beaulieu, V. Argiro, H. Ringl, A. Norbash, J. Feller, M. Dake, R. Jeffrey, and S. Napel. Perspective Volume Rendering of CT and MR Images: Application for Endoscopic Imaging. In *Radiology*, volume 199, pages 321–330, 1994.
15. N. Scott, D. Olsen, and E. Gannett. An Overview of the VISUALIZE fx Graphics Accelerator Hardware. *The Hewlett-Packard Journal*, (May):28–34, 1998.
16. L. Serra, W. Nowinski, T. Poston, N. Hern, L. Meng, C. Guan, and P. Pillay. The Brain Bench: Virtual Tools for Stereotactic Frame Surgery. In *Medical Image Analysis*, volume 1(4), pages 317–329, 1997.
17. M. Vannier, J. Marsh, and O. Warren. Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation. In *Proc. of ACM SIGGRAPH*, pages 263–273, 1983.
18. D. Vining, D. Gelfand, R. Bechtold, E. Scharling, E. Grishaw, and R. Shifrin. Technical Feasibility of Colon Imaging with Helical CT and Virtual Reality. In *Annual Meeting of American Roentgen Society*, page 104, 1994.
19. H. Zhang, D. Manocha, T. Hudson, and K. Hoff. Visibility Culling Using Hierarchical Occlusion Maps. In *Proc. of ACM SIGGRAPH*, pages 77–88, 1997.

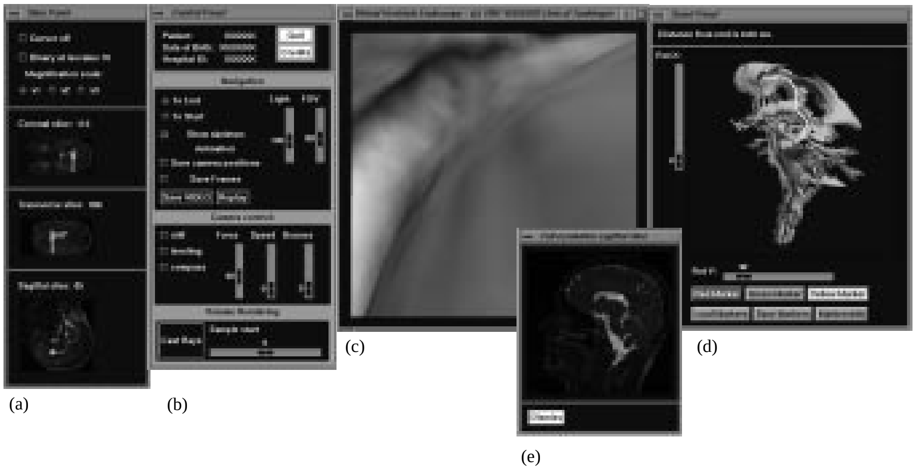
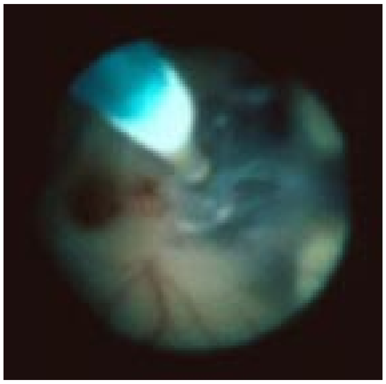
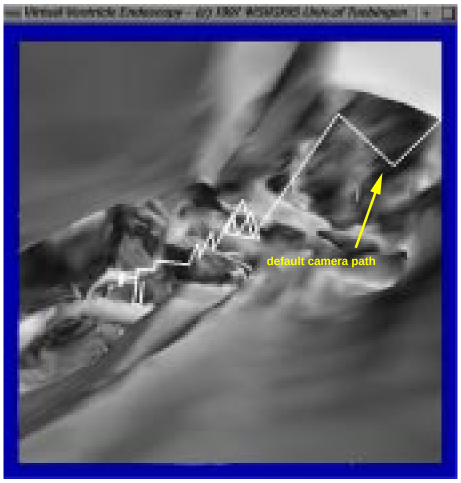


Fig. 8. User-interface of the virtual ventriculoscopy system: (a) Slice Panel, (b) Control Panel, (c) Main View, (d) Scout Panel, and (e) Full Resolution Sagittal Slice Panel.

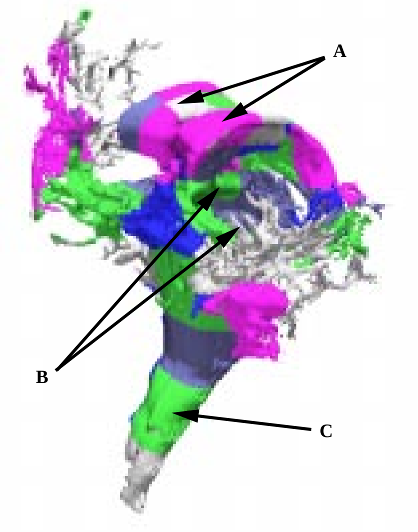


View through optical endoscope. (Bartz et al., Fig. 2)

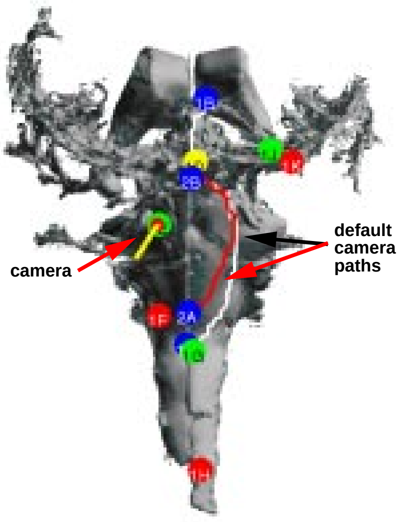


Virtual Ventricle Endoscopy - (c) 1997 WSI/GRIS Univ.of Tuebingen
default camera path

View through virtual endoscope: The voxel-based default camera path is rendered as white line. (Bartz et al., Fig. 5)



Octree Subdivision of Ventricular system - different blocks are marked with different colors. A: Lateral Ventricles, B: 3rd and 4th Ventricles, C: Central Canal. (Bartz et al., Fig. 6)



Snapshot from Scout Panel with front-view of ventricular system: Two default camera paths are shown (white and red); the pre-defined markers are colored in blue. (Bartz et al., Fig. 7)