Epileptogenic Foci Visualizator: An interactive computer-aided diagnosis tool

Cleber Roberto Milani[1], Gerson Geraldo H. Cavalheiro[1], and Lucas Ferrari de Oliveira[3]

[1] Universidade Federal de Pelotas - UFPel
96010-900 – Pelotas, RS – Brasil
`cleber.milani@pucrs.br`
`gerson.cavalheiro@ufpel.edu.br`
[2] Setor Escola Técnica da Universidade Federal do Paraná – UFPR
81520-260 – Curitiba, PR – Brasil
`lferrarioliveira@gmail.com`

Abstract. This paper presents Epileptogenic Foci Visualizator - EFV, a tool developed to assist pre-surgical evaluation of intractable epilepsy patients. The software implements a volumetric reconstruction algorithm to allow the visualization of multiple medical data through an interactive manipulation interface. Aiming processing time reduction to enable EFV satisfactory interactivity in low-cost personal computers, its implementation employs multithreading programming with POSIX threads in multi-core processors. We have obtained up to 25% of reduction at execution time considering different dual-core configurations.

Key words: Computer aided diagnosis (CAD) – Volumetric Visualization – Medical Image – Epilepsy

1 Introduction

Epileptogenic Foci Visualizator (EFV) is a tool developed by [1] to assist diagnosis. Its goal is to support professionals of the medical area on the evaluation of patients with intractable epilepsy. It is estimated that Epilepsy is a disease present at the daily lives of approximately 50 million people around the world and that only 25 to 45% of them are seizure free after 12 months with Antiepileptic drugs (AED) treatment [2]. Remaining cases are classified as intractable and thus require surgical intervention. Surgical intervention requires a previous evaluation that can benefit from the complementary information obtained from two or more nuclear medicine or radiology images, of equal or different modalities [3].

A traditional process to evaluate intractable epilepsy patients is visual side-by-side comparison of SPECT (Single Photon Emission Computed Tomography) images taken when the patient is in crisis with others taken during the crisis intervals. These images are functional because they present information related to the metabolic activity and are called, respectively, ictal or critical and periictal or inter-critical images. The goal when comparing these images is to identify which parts of the brain become active specifically during an epileptic crisis, aiming to identify the epileptogenic focus. However this technique when manually applied, besides being a laborious and exhaustive work, brings some limitations to the reproducibility of the studies since it depends on previous experiences of the investigator and the criteria adopted during the evaluation, which can very often be subjective.

Along the years, various computational tools have been developed to automate and assist pre-surgical evaluations. The SISCOM (Subtraction Ictal SPECT Co-registered to MRI) method is very often used for epilepsy evaluation. This method allows the detection of the epileptic focus from the subtraction of SPECT functional images and the anatomical localization of this focus at the Magnetic Resonance exam (MR) [4]. MR images can give information related to the structures of the observed organs and are, for this reason, classified as anatomical images.

In [2, 3, 5] the authors proposed the technique called B.R.A.S.I.L. based on SISCOM methodology but that use fusion of images in three dimension (3D) to produce functional images aligned to the anatomical images referent to the portions at the Coronal (axis X), Axial (axis Y) and Sagittal (axis Z) positions inside the patient brain. The fusion of images aims to group complementary information of the different exams, which would usually be visualized separately. The authors also discuss the importance of the alignment and the methodology developed as well as the origin of the images. This technique represents an advancement in the sense that visualization of 3D images aligned is very useful for the qualitative analysis during the decision making phase.

However, in [3] the author did not develop any tool that allows the visualization, processing and interaction with the images resulting from the proposed method. Commercial tools that support medical images visualization, such as VolView [6], besides having a very high cost, do not allow the presentation of the final result of the fusion and alignment proposed. Considering this context, the project described in this paper has been developed aiming at offer an application for interactive 3D visualization of the fusion and alignment of images proposed by [3]. This way it is possible to offer a visualization of the problem very close to the real one, with the support of the colors to highlight the epileptogenic focus. In this sense, the main contribution of this project is the development of a tool to deal with the problem of simultaneous evaluating multiple complementary medical images of different origins using the methodology proposed by [3].

The rest of this paper is organized as follows: sections 2 and 3 present, respectively, the requirements identified for the development of the software proposed and the implementation of the tool itself. Section 4 describes the results of the tool evaluation. Section 5 concludes the paper.

2 Project Requirements

Various requirements have been identified for the development and implementation of the project proposed. The first of them refers to the visualization of data. In general, the acquirement of volumetric data in medicine produces images composed of parallel slices uniformly separated. The volume (tridimensional image visualization) is then generated through the piling up of these slices maintaining the original space between them and mapping pixels into voxels. A voxel is one volume's element that represents values in tridimensional space, analogous to a pixel in 2D images. This is done through the technique of volumetric reconstruction, which uses a Ray-tracing algorithm and allows 3D interactive visualization of the images in a realistic way [7]. In order to meet this requirement, the VTK (Visualization ToolKit)[8] implementation of Ray-casting, which is a simplified version of Ray-tracing, was used in this project.

Additionally, the Image registration is much used to improve the sensitivity and specificity of complementary procedures, with the aim of detecting, locating, monitoring, and measuring pathological and physical disorders. Often, registration contributes by bringing new or extra information, leading to the development of a great number of registration algorithms where the corresponding pixel coordinates on different images are transformed in order to align and match its position and spatial coordinates. The ictal and periictal SPECT images aligned and subtracted is the most used technique to determine epileptogenic zone location [5]

Another requirement identified was the elaboration of a friendly interface that allows the user to change at execution time the parameters applied to the exams reconstruction. In Ray-casting, the appearance of a voxel in the screen is determined by a calculation composed of a series of factors, among which it is possible to highlight the transfer functions. The goal of these functions is to map the information of a voxel in different values of color and opacity. Therefore the dynamical modification of the transfer functions (among other parameters) is indispensable, since the set of options that allows a good visualization of the exam of a certain patient may not be the most suitable one for a different case. Besides, the possibility of dynamical change offers the user different visions of the same situation, which can eventually facilitate the observation of less evident details.

Interactive visualization applications require a reconstruction frame rate between 10 and 20 frames per second to be considered as a satisfactory interaction with the user [9]. Thus, the existence of a user interface introduces a new concern to be dealt with: the computational cost. Special attention must be paid to the very high computational cost of the Ray-casting algorithms, which complexity is $O(n^3)$[7]. By contrast, the algorithm can be parallelized since it determines the pixel values through the launching of rays, which are independent from each other. Therefore, the algorithm can be explored in parallel architectures with performance gains. According to [10], Ray-casting of large data sets is not possible at interactive frame rates unless massive parallel hardware is used.

In fact, literature presents some proposal of solutions to the high computational cost problem of medical images visualization using parallel programming resources. In [11] is presented an approach to image visualization on computer clusters that employs MPI (Message Passing Interface)[12] library for parallel programing. In [13] is presented a framework to visualize multiple volumes, which uses the calculation potential offered by GPUs (Graphics Processing Unit).

However, looking at commercial context, it is possible to observe that, lately, powerful parallel computers with multi-core processors have been considered as commodities. This type of parallel architectures has become accessible to hospitals, clinics, private offices and even to doctors themselves. Thus, considering its availability among the target public, the development of new parallel applications and the optimization of already existing applications for these architectures using parallelism is a high need. It is also an interesting fact that many multithread programming libraries are available for free to facilitate the exploration of parallelism in multi-core processors. Therefore, to explore the parallelism inherent to the Ray-casting algorithm using multithread programming in multi-core computers was considered the most interesting and viable solution for the problem identified.

The social impact involved in this kind of software was considered as a final requirement to the project. The principle adopted was making the tool accessible to the highest number of users possible at the lowest possible cost. By this we refer specially to researchers that does not have access to proprietary softwares or specific hardware architectures. Three directions were considered to meet this requirement: free software based development; development focused on architectures with accessible price for potential users; implementation that would result in a software without special hardware requirement or additional software dependencies to execute successfully.

Concerning the free software item, besides the fact that EFV is to be distributed as free software[3]; its development has used libraries and applications available under the same principles. As a consequence, besides being free of charges the tool offered can be modified to meet local needs of the users, which does not happen with proprietary software, such as VolView, which license costs US$ 2.500,00 and which is not open source. Considering the architecture focus, besides the commercial context described above, EFV has been developed in a portable way, i.e., it can be executed without restrictions on most of the general purposes computers. Finally the implementation of a software without special hardware and software requirements allows EFV to operate satisfactorily in operating systems such as Windows XP, Linux and Mac OS X, as well as in computers without a 3D graphics card.

3 EFV Implementation

EFV was implemented in C++ and the POSIX Threads (pthreads) specified by IEEE Standard 1003.1c-1995 was adopted as support to multithread program-

---

[3] Available at http://epileptogenicsf.sourceforge.net/

ming. The images format used was the Analyze 7.5 [14]. The libraries VTK [8] and FLTK (Fast Light ToolKIT) [15] have been used respectively to describe the visualization algorithms and build the user interface.

EFV offers two ways to visualize the epileptogenic foci: the first one through the reconstructed volumes and the other one, auxiliary, through bidimensional slices. The main window of the application (Figure 1) consists in three scenes: the leftmost frame shows the reconstruction of the patient skin, the rightmost frame shows the epileptogenic foci isolated and the central frame shows the result of the alignment and fusion of the exams processed, i.e., the epileptogenic foci anatomically located. The interaction with the volumes that form the images can be separated or simultaneous, which can be defined by the user at the bottom right quadrant simply by indicating which volumes must suffer the influence of the interaction. The visualization in bidimensional slices also shows three scenes: one for the Axial axis, another one for the Coronal axis and the last one relative to the Sagittal axis of the brain (Figure 2). It is also possible to choose the language of EFV at the menu between English-EUA and Portuguese-BR.

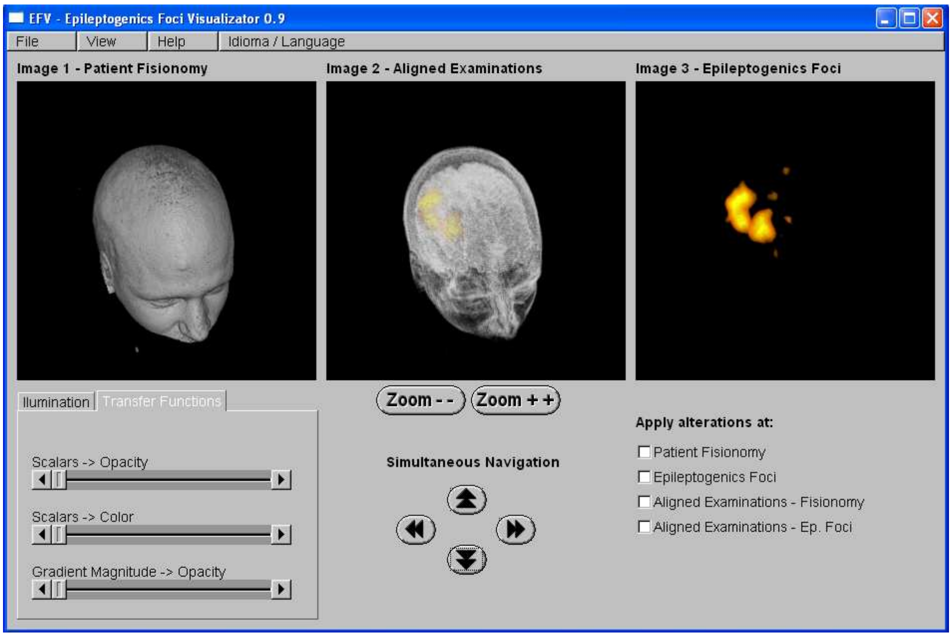
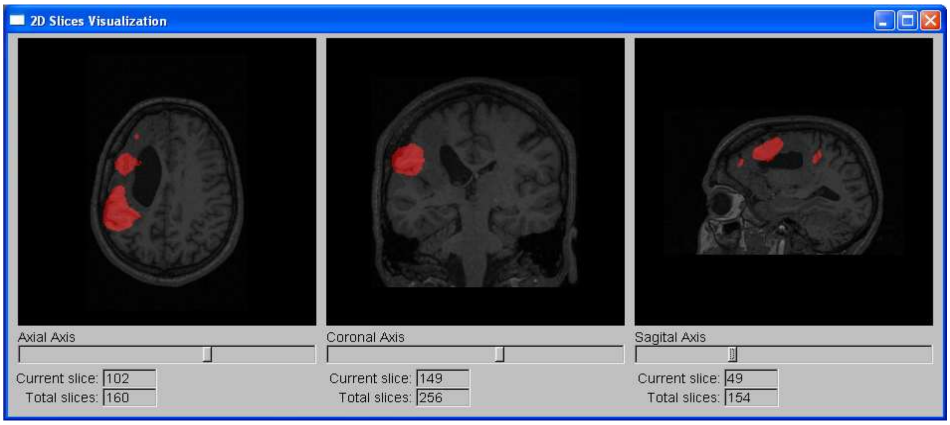


Fig. 1. EFV main window.

Ray-casting is implemented by VTK through a visualization pipeline on which various properties can be defined for the reconstruction of images. As described in the previous section, the amount of information present in an exam is very large. Therefore, offering the user the possibility to vary some of the



Current slice: 102
Total slices: 160

Current slice: 149
Total slices: 256

Current slice: 49
Total slices: 154

Fig. 2. Visualization in bidimensional slices of the aligned exams.

parameters at execution time facilitates the interpretation of exams. However, some of these properties, such as Ray-casting algorithm and the interpolation type, are not interesting to target public of EFV. Additionally, the definition of most appropriate transfer functions parameters is a difficult task that requires constant research efforts. Literature presents works that apply constrained non-linear optimization methods, Fourier Transform, among others [16] to determine these parameters. Therefore, it has been decided to maintain the Ray-casting function and the interpolation type constant leaving only some previously defined illumination and transfer functions available for user manipulation.

There are also some performance related parameters defined as manageable during execution time. The first one is the number of threads used at the reconstruction of each volume; the second one is the number of frames per second expected during the interactions of the user with the volumes (rotate an image, for example). These parameters can be changed at the menu view/advanced options. The configuration of the number of threads aims to assure EFV software scalability, because when the number of processors is increased on the computer where the application is executed, the software can be adapted to take advantage of this new number. In order to change the expected frames per second rate, it is necessary to call internal VTK routines that deal with the level of detail used to show the volumes. In case the rate defined by the user is higher than the one the computer could generate, the volumes will be presented with a lower level of detail while the user executes the interactive operations and at the end of the interaction they will be presented again at their usual quality.

4 Evaluation and Results

The evaluation of the tool has considered the performance gains obtained with the use of thread resources and the quality of the images built. Furthermore, the application has been compiled in various operating systems in order to verify its portability. Visualization results has been evaluated by qualitative comparison of individual images generated by EFV with those presented by the proprietary software VolView. Many evaluations have been accomplished concerning performance, but considering the space available in this paper, only the main evaluation will be discussed here. Additional tests and a complete documentation of the results as well as the description of the experiments is available at [1].

Performance evaluations employed the *GetTimeToDraw()* method from *vtkVolumeRayCastMapper* class to compute Ray-casting execution time. Additionally, *vtkTimerLog* class was used to compute total time and time spent on I/O operations. The number of samples used in the tests was 100. This means that every time a reconstruction parameter was changed, the volumes were reconstructed and reloaded a hundred times. The upper and lower bounds, i.e., highest and lowest execution times, was removed and the final result was obtained by calculating the arithmetic mean of remaining times. This value was chosen because it represents a significant sample size, which increases results reliability by minimizing weight of external influences such as system interruptions.

Table 1 presents the configurations of the architectures used at the performance tests. During the execution of these tests, architectures A1 and A7 executed Windows XP OS and architecture A8 executed Mac OS X 10.4.10. With Linux OS only portability evaluation was accomplished and not performance evaluation. In order to facilitate the comparison of the execution in different hardware and software environments, all the compilations were performed without any optimization parameter.

Table 1. Configurations of the architectures used at the tests phase.

Architecture	**Processor**	**Cores**	**L2 Cache**	**RAM Mem.**	**VGA**
A1	AMD Semprom 2600+ @ 1.84 GHz	1	256 KB	1 x 512 MB 333 MHz	Cirrus Logic 2 MB
A2	AMD AthlonXP 2000+ @ 1.67 GHz	1	256 KB	2 x 256 MB 266 MHz	GeForce 5200 128MB
A3	Intel Pentium 4 HT @ 3.00 GHz	1 HT	1 MB	1 x 512 MB 333 MHz	Shared 32MB
A4	Intel Pentium D 805 @ 2.66 GHz	2	2 x 1 MB	1 x 512 MB 333 MHz	Shared 16MB
A5	Intel Core 2 Duo E6300 @ 1.86 GHz	2	2 MB	2 x 1 GB 667 MHz	GeForce 7600GS 256MB
A6	Intel Core Duo T2300 @ 1.66 GHz	2	2 MB	2 x 512 MB 667 MHz	Shared 128MB
A7	AMD Turion 64 X2 TL-50 @ 1.60 GHz	2	2 x 256 KB	1 x 1024 MB 667 MHz	Shared 128MB
A8	PowerPC G4 @ 1.5 GHz	1	512 KB	2 x 512 MB 333 MHz	ATI Mobile Radeon

Table 2 presents the total time spent in the reconstruction of the three volumes that are presented in the tool, varying only the number of threads in each reconstruction. Columns V1, V2 and V3 represent the number of threads used in the reconstruction of the skin volumes, aligned exams and epileptogenic foci, respectively. For example, V1=V2=V3=1 means that three threads have been executed, one after the other, while V1=V2=V3=2 indicates the execution of six threads, two at each time. Columns A1, A2 etc, refer to the configurations defined in Table 1. The time presented in the table represents how long the user would effectively have to wait after requesting the loading of the exams to visualize at the tool, including the time consumed by input and output operations. Even though these operations are not constantly performed during the interactive process (such as file reading, for example) all of them must be executed at the moment of the first reconstruction.

Table 2. Average time spent in the reconstruction of the three volumes in various architectures, including I/O operations.

Threads			**Total reconstruction time in seconds**							
V1	**V2**	**V3**	**A1**	**A2**	**A3**	**A4**	**A5**	**A6**	**A7**	**A8**
1	1	1	6,557	7,182	5,475	5,308	3,037	3,919	5,237	21,201
1	1	2	6,550	7,196	5,447	5,052	2,861	3,662	4,897	21,298
1	2	1	6,528	7,220	5,316	4,721	2,689	3,426	4,600	21,363
1	2	2	6,534	7,224	5,284	4,467	2,510	3,175	4,248	21,336
2	1	1	6,524	7,231	5,378	5,030	2,891	3,709	4,911	21,355
2	1	2	6,542	7,239	5,356	4,785	2,707	3,457	4,592	21,345
2	2	1	6,554	7,278	5,232	4,452	2,536	3,222	4,287	21,357
2	2	2	6,553	7,310	5,191	4,205	2,361	2,980	3,965	21,003
4	4	4	6,580	7,314	5,206	4,187	2,413	3,052	4,038	21,061
8	8	8	6,586	7,439	5,200	4,203	2,408	3,050	4,008	21,226

In multi-core architectures in general the most satisfactory result was obtained when reconstructing the three volumes with 2 threads each. The A4 architecture was the only one not to follow the trend, obtaining better results using 4 threads per volume. The reduction of processing time, comparing the execution with 1 thread with the multithread one was, in this case, of approximately 21% and the speedup of 1,27. However, it is possible to observe that the gains with the use of 2 threads per window were very similar. Architecture A8 has presented the less satisfactory performance of the group. When comparing the lowest processing time spent by a dual-core architecture (architecture 5 with 2 threads per volume) with the average time necessary for the reconstruction at single-core architecture A1, it is possible to notice a reduction of 64% of processing time. The results obtained in dual-core portable computers have also reached the expectations. In architecture A6, the reduction of processing time was of 24% with a speedup of 1.31 and an efficiency of 65%. On the other hand, architecture A7, has presented a reduction of 25% at processing time, speedup of 1.32 and an efficiency of 66%.

Finally, as expected, the comparison between the performance of architectures A1 and A2 indicates that the presence of hardware for 3D graphics acceleration does not interfere on the tool performance. GPUs are employed by applications on which surface rendering algorithms, such as Dividing Cubes and Marching Cubes, are used to create the volumetric visualization. However, when Ray-casting algorithms are employed the applications in general does not take advantages from GPUs. This was a expected and also a satisfactory result since it confirms that the goal to develop a software without special graphics hardware requirements has been achieved.

5 Conclusion

EFV has presented satisfactory results concerning portability, performance, scalability and the visual result generated. Hardware portability has been proved with the execution of the tool in single-core and multi-core architectures of various manufacturers. The sample has included processors that, besides having different numbers of cores, also have different sets of instructions. Furthermore, 32 and 64 bits processors have been both successfully tested. Software portability was proved by successfully compiling and executing the tool at the operating systems Windows XP, Debian Linux and Mac OS X 10.4.10.

The exploration of the parallelism inherent to the Ray-casting algorithm, dividing it in independent execution flows, has resulted in some performance gain on architectures having processors with the Hyper Threading technology and in multi-core processors. On the other hand, on single-core processor architectures, the execution time variation was irrelevant. The frames per second rate defined as suitable for interactive applications is achieved at EFV by applying, together with the parallelism, the technique that allows the reconstruction of the volume with a variable level of details during the interaction. This rate can still be changed at execution time in the user interface.

Software scalability is also a characteristic present in the tool developed. Because the tool allows dynamically manipulating the number of threads during execution time in case the number of processors of the architectures is increased, it is possible to configure the software to efficiently explore these processors through the division of Ray-casting in a higher number of execution flows.

Finally the qualitative comparison with the proprietary software VolView indicates that the final result of the project was also satisfactory. Besides the fact that the reconstructed exams present similar quality, EFV also offers the functionalities of VolView that are of interest on epilepsy evaluation, such as the visualization of exams in 3D volumes or 2D slices and the possibility of interact and dynamically change the volumetric visualization.

The goal to develop a tool for visualization of 3D images fusion has been therefore achieved in a satisfactory way. At the conclusion of this project it is available a computational tool which efficiently implements a new visualization approach to assist the detection of the epileptogenic foci.

References

1. Milani, C.R.: Ferramenta multithread para reconstrução volumétrica de imagens em medicina nuclear para auxílio na detecção de zonas epileptogênicas. Undergraduate conclusion work, Instituto de Física e Matemática, Universidade Federal de Pelotas, Pelotas (2007) 129p.
2. Oliveira, L.F., Azevedo-Marques, P.M., Wichert-Ana, L., Sakamoto, A.C.: Brain registration and subtraction - improved localization for spect analysis (B.R.A.S.I.L.): a computer-aided diagnosis in epilepsy tool kit. In: SAC '08: Proceedings of the 2008 ACM symposium on Applied computing, New York, NY, USA, ACM (2008) 1390–1394
3. Oliveira, L.F.: Fusão de Imagens 3D para Localização e Quantificação de Zona Epileptogênica. PhD thesis, Faculdade de Medicina de Ribeirão Preto, Universidade de São Paulo, Ribeirão Preto (2005) 151p.
4. O' Brien, T., So, E., Mullan, B., Hause, M.and Brinkmann, B., Bohnen, N., Hanson, D., Cascino, G., Jack, C.J., Sharbrough, F.: Subtraction ictal spect co-registered to mri improves clinical usefulness of spect in localizing the surgical seizure focus. Neurology **50** (1998) 445–454
5. Oliveira, L.F., Azevedo-Marques, P.M., Ana, L.W., Sakamoto, A.C.: Implementation and test of B.R.A.S.I.L: An epilepsy computer-aided diagnosis toolkit. In: CSE '08: Proceedings of the 11th IEEE International Conference on Computational Science and Engineering. (2008) 227–233
6. Kitware VolView, Interactive and Intuitive Volume Visualization http://www.kitware.com/products/volview. (2009)
7. Schroeder, W., Martin, K., Lorensen, B.: The Visualization Toolkit - An Object Oriented Approach to 3D Graphics. 3rd edn. Kitware Inc. (2002)
8. The Visualization Tool Kit http:// public.kitware.com/VTK. (2009)
9. Robb, R.A.: Visualization in biomedical computing. Parallel Computing **25** (1999) 2067–2110
10. Bajaj, C., Ihm, I., Koo, G.B., Park, S.: Parallel ray casting of visible human on distributed memory architectures. In: Data Visualization 99, Eurographics (1999) 269–276
11. Manssour, I.H., Fernandes, L.G., Freitas, C.M.D.S., Serra, G., Nunes, T.: High performance approach for inner structures visualisation in medical data. IJCAT **22** (2005) 23–33
12. Snir, M., Otto, S., Huss-Lederman, S., Walker, D.W., Dongarra, J.J.: MPI: The Complete Reference. MIT Press, Cambridge, Massachusetts, USA (1996)
13. Roβler, F., Tejada, E., Fangmeier, T., Ertl, T., Knauff, M.: Gpu-based multivolume rendering for the visualization of functional brain images. In: Proceedings of SimVis 2006. (2006) 305–318
14. Biomedical Imaging Resource http://www.mayo.edu/bir/PDF/ANALYZE75.pdf: AnalyzeTM 7.5 File Format. (2007)
15. Fast Light Toolkit http://www.fltk.org. (2009)
16. Potts, S., Möller, T.: Transfer functions on a logarithmic scale for volume rendering. In: GI '04: Proceedings of Graphics Interface 2004, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, Canadian Human-Computer Communications Society (2004) 57–63