

Interactive Diffusion Based Smoothing and Segmentation of Volumetric Datasets on Graphics Hardware

J. Beyer¹, C. Langer¹, L. Fritz¹, M. Hadwiger¹, S. Wolfsberger², and K. Bühler¹

¹ VRVis Research Center for Virtual Reality and Visualization, Vienna, Austria

² Department of Neurosurgery, Medical University Vienna, Austria

Contact:

Johanna Beyer, VRVis Research Center, Donau City Str. 1, 1220 Vienna

Tel: +43(1)20501-30703, Fax: +43(1)20501 30900, Email: beyer@vrvis.at

Summary. Objective: Volume segmentation with concurrent visualization is becoming an increasingly important part of medical diagnostics. This is due to the fact that the immediate visual feedback speeds up evaluation of the segmentation process, hence enhances segmentation quality. Therefore, our aim was to develop a method for volume segmentation and smoothing which achieves interactive performance on standard PCs and is useful in clinical practice (i.e. fast and of high-quality).
Methods: Our application is based on seeded region growing and nonlinear isotropic as well as anisotropic diffusion. We use current GPUs (graphics processing units) to speed up the computation of the diffusion process and use hardware accelerated interactive volume rendering.
Results: Using our approach the user can observe the diffusion process in real-time, change parameters interactively and view the result in a high-quality 3D direct volume rendering (DVR).
Conclusion: The interactive nature of our algorithm and simultaneous visualization improved the usability of our segmentation and smoothing algorithm and proved useful in the clinical workflow. Using our application we were able to speed up the (an)isotropic diffusion process to achieve interactive performance.

Keywords. Volume Segmentation, Volume Smoothing, GPU, Diffusion, Direct Volume Rendering

1 Introduction

In today’s clinical practice, computer aided diagnosis and surgery planning applications are getting increasingly important. These specialized applications, which rely on high-resolution 3D images (e.g. CT or MRI scans), often require integrated volume segmentation possibilities. Already, a huge amount of medical image segmentation solutions exists, however, processing time, reliability, interactivity, high-quality realtime visualization and ease of use are still ongoing issues.

Image smoothing for medical applications is mainly needed for the enhancement of image quality or as a preprocessing step prior to segmentation. The aim is to reduce noise while preserving important features which could improve the segmentation result.

Segmentation methods can be divided into manual, automatic and semi-automatic approaches. Manual approaches are very time consuming, however, for a lack of better alternatives they are still common in clinical practice. Automatic approaches, on the other hand, are often very limited in their scope of application because of their highly specialized algorithms. Additionally, adjustments of global parameters may require costly recalculations.

We propose a GPU-based algorithm which applies nonlinear isotropic and anisotropic diffusion filtering for image smoothing, as well as seeded region growing combined with diffusion for segmentation. The algorithm shows the evolution of the diffusion process in real-time. Our approach is fast, semi-automatic, and does not require costly recalculations after changing parameter settings. Real-time visualization is achieved by hardware accelerated DVR (direct volume rendering) which allows the user to interact with the ongoing segmentation.

1.1 Related Work

The idea of diffusion filtering comes from multiscale image analysis with the goal to reduce noise while minimizing information loss. Smoothing of medical images by diffusion is well covered in literature [1, 2]. The usual application of diffusion filtering (i.e. smoothing and edge enhancement) was extended by Sherbondy et al. [3] to segmentation purposes by utilizing GPUs (graphics processing units) for accelerating the computation process. GPGPU (general purpose computing on GPUs) exploits the highly parallel structure of GPUs to speed up otherwise computationally expensive algorithms such as solving partial differential equations [4]. Even though early approaches of utilizing GPUs for 2D diffusion filtering [5] or 3D convolution [6] have suffered from limited precision, they could show the increased computational power compared to CPUs. Sherbondy et al. were the first to introduce diffusion-based segmentation on GPUs, however, their work falls short in several areas. First, they only implemented the basic Perona and Malik nonlinear diffusion scheme [7] and secondly, they use a very rudimentary visualization without the possibility to arbitrarily change rendering parameters or transfer functions. Also, they only render the segmented object without the surrounding volume. We, therefore, extended Sherbondy’s work by implementing full anisotropic edge-enhancing diffusion on GPUs and integrated the diffusion algorithm into a high-quality interactive visualization framework [8]. For more in-depth discussions about diffusion filtering the reader is referred to [1, 2, 9].

Our visualization framework is based on DVR [10], which creates images directly from volumetric data without the need to extract any geometry first. The major advantage of DVR is the increased amount of information that can be conveyed in one image by making use of transparency, which allows to peer inside the volume. Additionally, transfer functions and lighting can significantly enhance the 3D perception of the volumetric structure. Hardware accelerated approaches allowing interactive high quality volume visualization on standard PCs range from 2D texture mapping based algorithms [11] to more recent GPU-based raycasting [12]. Rendering of segmented volume data imposes the additional problem of filtering object boundaries at high resolution and has been addressed by Hadwiger et al. [8] who use GPU-based two-level volume rendering to specify transfer functions and render modes on a per object basis with trilinear object filtering.

2 Methods

Our smoothing and segmentation approach is based on nonlinear isotropic as well as edge-enhancing anisotropic diffusion filtering. Due to the high computational demands of (anisotropic) diffusion, interactivity can only be achieved by exploiting the computational power of programmable graphic cards. Therefore, we implemented nonlinear isotropic diffusion on GPUs and also extended the implementation of anisotropic diffusion to completely run on graphics hardware. For segmentation we employ a seed-based region growing algorithm while visualization is achieved by two-level volume rendering. Our main contribution lies in the integration of the GPU-based segmentation and smoothing algorithms into a high-quality interactive visualization environment. Such an interactive system not only requires a fast segmentation algorithm which can be adjusted interactively, but also a visualization system which can display the ongoing segmentation in real-time while offering a high quality volume rendering of the underlying data. Our implementation is based on C++, OpenGL and the Cg shading language. Figure 1 gives an overview of the segmentation and visualization processing pipeline.

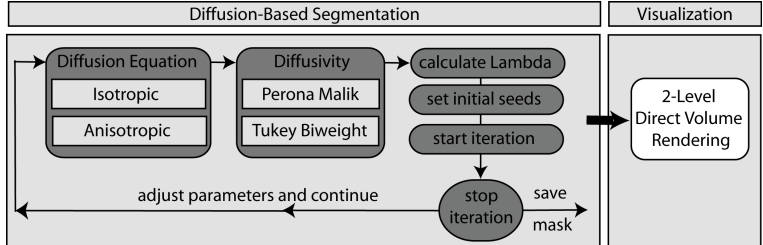


Figure 1. The segmentation processing pipeline.

2.1 Diffusion

The common notion of diffusion is a physical process that equilibrates concentration differences without creating or destroying mass. This can be applied for image smoothing where the amount of smoothing depends on the gradient’s direction and magnitude. We use nonlinear isotropic as well as anisotropic diffusion [1]. In the isotropic case, a diffusivity function $g()$ is used to control the intensity of the flux, whereas anisotropic diffusion uses a tensor D instead.

Nonlinear isotropic diffusion incorporates a feedback system which scales the diffusion based on the gradient of the evolving image (instead of using the gradient of the original image):

$$\partial_t u = \text{div} (g(\|\nabla u\|)\nabla u). \quad (1)$$

div denotes the divergence operator and ∇u the gradient of the data in the density volume (e.g. CT volume). $g(\|\nabla u\|)$ scales the gradient in order to stop the flux at edges. We employ the Perona Malik [7] (Equ. 2) and the Tukey Biweight model [13] (Equ. 3) for calculation of the diffusivity g :

$$g(s) = e^{-(s^2/\lambda^2)} \quad (2) \quad g(s) = \begin{cases} \frac{1}{2}[1 - (\frac{s}{\lambda})^2]^2, & |s| \leq \lambda \\ 0 & \text{else.} \end{cases} \quad (3)$$

Parameter λ scales the gradient, thus influences the edge detection. For discretization of the diffusion equation in 1 we use the formula presented in [7] but extended the neighborhood of six voxels to optionally eighteen- and twenty-six neighbors by additionally fetching the respective neighbor voxels. This larger neighborhood allows to segment smaller details and to further enhance edges during smoothing.

In *edge-enhancing anisotropic diffusion*, instead of using the scalar diffusivity $g()$, the tensor D is

used, which not only scales but also rotates the flux in order to preserve interesting features. The orthogonal system of eigenvectors v_1 , v_2 and v_3 of the diffusion tensor D for a three dimensional volume can be written as follows:

$$D = [v_1 \ v_2 \ v_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} v_1^\top \\ v_2^\top \\ v_3^\top \end{bmatrix} \quad (4)$$

where $v_1 \parallel \nabla u_\sigma$, $v_2 \perp \nabla u_\sigma$ and $v_3 \perp \nabla u_\sigma$. ∇u_σ is the gradient defined by using a Gaussian kernel with width σ . To obtain smoothing along the edges, the corresponding eigenvalues can be selected as $\lambda_2 = \lambda_3 = 1$. λ_1 is defined using one of the diffusivities in Equation 2 or 3. If $\lambda_1 = \lambda_2 = \lambda_3 = g()$ it results in the isotropic diffusion.

Our implementation needs two render passes in order to cache the result of the on-the-fly computation of smoothed gradients. The first pass calculates the gradients using a gaussian kernel. We achieved good results with a kernel of width 3 and a σ of 0.5. The second pass uses the diffusion stencils together with the gradient volume to calculate the divergence. Finally, the stencil is applied on the original density, added up and normalized.

2.2 Image Smoothing

We use the above described diffusion methods for noise removal, edge enhancement and as a preprocessing step prior to segmentation. Each iteration of the diffusion process is visualized in the volume rendered 3D view as well as the 2D MPR (multi-planar reconstruction) views, so that the user can stop the diffusion process at the desired level of smoothness. Additionally, a starting value for the parameter λ can be calculated due to the fact that for the flow in image smoothing, the proportion between $\|\nabla u\|$ and λ stays the same in all scales. ∇u is calculated by building the difference of two user specified density values from each side of the edge that is to be enhanced. Tukey flow stops exactly when the parameter λ is as large as the image gradient whereas the Perona Malik flow approaches zero very closely at $\|\nabla u\| = 3\lambda$.

2.3 Segmentation

Generally, in region-based techniques seeds evolve in each iteration of the algorithm as long as a certain homogeneity criterion is satisfied. Using a hybrid of seeded region growing and diffusion filtering, seed evolution is controlled by partial differential equations. In contrast to image smoothing, segmentation applies the diffusion equation on the seed volume, not directly on the density volume. Diffusion of the seeds, however, is defined by the data in the density volume. After each iteration the object membership of all voxels in the original volume is updated and visualized. Our seeded region growing segmentation algorithm consists of several parts:

- 1. Seed Selection:** The user draws seed points for the segmentation process into the orthogonal slice views, which indicate the desired region to be segmented. These seed points are saved in a seed volume, a secondary volume in addition to the density volume.
- 2. Segmentation:** The merging criteria of the region growing process is based on diffusion filtering, where the diffusion flow depends on the density volume but is multiplied with the gradient magnitude of the seed volume. The segmentation process continues (i.e. seeds progress along the flux) until the expansion is stopped at the edges.
- 3. Computational Masking:** To optimize performance, voxels that do not influence the diffusion process are culled by a depth test before starting the computation. As the diffusion process only takes place in the neighborhood of a seed point, voxels are eliminated by dilating the seed mask by one and removing all voxels outside of it as described in [3].

2.4 Visualization

The progress of the segmentation and smoothing process is visualized in real-time in the context of the whole volume (i.e. original volume and the segmented objects). For rendering of segmented volume data we implemented hardware accelerated two-level volume rendering based on slicing as well as on GPU raycasting. Details can be found in our previous work [8, 14]. Two-level rendering allows to render segmented objects (e.g. brain, skull) using different render modes and transfer functions for individual segmentation masks. The user can, for example, combine unshaded DVR for high-detail rendering of the brain and shaded DVR of the skull for an improved 3D perception of the bone.

A great advantage of our interactive diffusion and visualization framework is its increased flexibility compared to other high-level segmentation approaches. Presenting only the final segmentation result prevents the user from fully understanding the segmentation process and makes it impossible to adjust parameters on the fly. Our approach allows the user to observe the running segmentation and intervene whenever necessary.

2.5 Interaction

During the segmentation process the user can, at any time, interact with the ongoing segmentation by stopping the iteration, deleting or adding new seed points, deleting erroneous segmentation marks, changing diffusion parameters, and continuing the iteration process. An example segmentation workflow is shown in Figure 2, a segmentation of the frontal and paranasal sinuses from a CT head dataset (256x256x128). The user starts with drawing seed points in the slice view (top



Figure 2. Interactive visualization of the ongoing segmentation.

left). After setting the parameters the segmentation process is started. The ongoing segmentation can be traced in the slice view (bottom left) as well as the 3D volume rendering (middle). The segmentation can be stopped by the user at any time, after which the user can delete erroneous seed points or change the diffusion parameters. Afterwards the user can continue the diffusion process until the segmentation is complete (right). During all these steps the user can also interactively adjust the volume rendering parameters (e.g. transfer function, render mode, lighting parameters). By combining interactivity, segmentation and visualization it is possible to create an application well suited for clinical practice.

3 Results

The main advantage of our segmentation algorithm is the interactivity provided by the fast, hardware-based implementation. We sped up the (an)isotropic nonlinear diffusion process by fully

relocating the calculations to the GPU. Additionally we integrated the diffusion methods into a high-quality visualization framework. Due to the immediate visual feedback the user can observe the diffusion process and interactively adjust the few existent parameters, which makes the approach feasible for a huge amount of different datasets.

3.1 Time Consumption

The segmentation algorithm has been tested on different medical datasets for segmentation of the colon, bones, vessels and the heart. We used an Intel Pentium IV 3.4GHz, 1 GB RAM with an ATI Radeon X800 graphics card. For segmenting the CT head dataset (256x256x128) we started a nonlinear isotropic diffusion process (Perona Malik) for smoothing, which was stopped after 30 iterations and took about 1.74 seconds. For segmentation of the paranasal sinuses we needed about 200 iterations (11.6 seconds) using the same diffusion equation. This leads to a cycle time of 0.058 seconds per iteration for isotropic diffusion and 0.51 seconds for an anisotropic diffusion step. To compare our smoothing algorithm to a software implementation of 3D Perona Malik nonlinear isotropic diffusion we used MATITK [15], a Matlab wrapper for ITK. The software version took about 9 seconds for one iteration step, showing the major speed improvement of a GPU implementation (155 times faster).

3.2 Quality Evaluation

We segmented a brain from an MRI using the Perona Malik equation. Our result was compared to a brain annotated by an expert. Our algorithm performed well, however, the setting of the parameters was a challenging task. Optimal automatic parameter setting, which can improve the segmentation quality a lot, is a research area on its own [16] and will be considered in our future work. The rightmost image in Figure 3 displays the correct segmentation results in gray and the difference between both methods in white. A more detailed evaluation of the quality of diffusion based segmentation is out of scope of this work.

3.3 Application

We integrated our diffusion-based segmentation and smoothing algorithm into an application for neurosurgeons which supports the workflow of presurgical planning by offering improved volume visualization of head data. The first step consists of smoothing of MR brain images using edge-preserving anisotropic diffusion. Next, we perform skull stripping by using our segmentation method (e.g. Tukey Biweight isotropic diffusion using a neighborhood of 18) and optionally segment additional structures of interest. Finally, by using our two-level GPU raycaster we can visualize anatomical details and improve their three dimensional perception. This integrated workflow considerably increases usability and is well perceived by the medical doctors. One example for a successful use of our diffusion application is the improved detection and visualization of focal cortical dysplasias (FCD) which are lesions of the cerebral cortex and a common cause for epilepsy. Results of our neurosurgical planning application can be seen in Figure 3.

3.4 Diffusion Methods Comparison

Evaluating the different diffusion methods, edge-enhancing anisotropic diffusion was found superior in regards to noise reduction in homogeneous regions (e.g. the brain in MRIs) and in preserving object contours. The drawback, however, was its long computation time compared to isotropic diffusion. The Perona Malik equation for isotropic diffusion took much less computation time with the drawback of blurring at small discontinuities while the Tukey Biweight model maintained sharper boundaries and needed in general fewer iterations.

The extension of the neighborhood for the diffusion equation (see Section 2.1) has the advantage of being able to segment smaller details due to the propagation of seeds into the additional

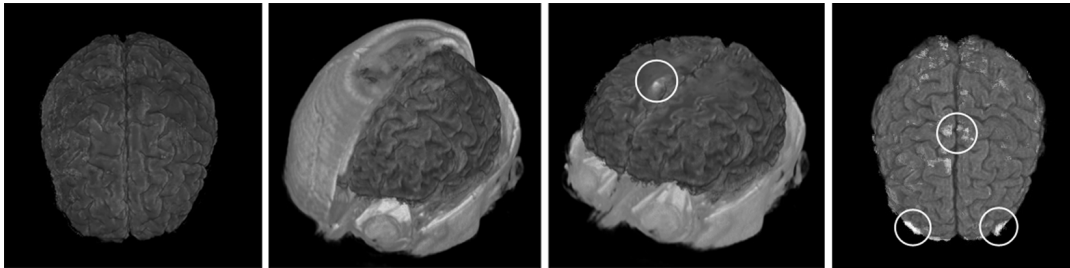


Figure 3. Left: Segmented and extracted brain. Left center: Two-level volume rendering showing the segmented brain and skull. Right center: Improved visualization of FCDs. Right: Segmentation error.

neighbors. The drawbacks are a longer computation time and possible leaking at weak contours. For smoothing, a larger neighborhood enhances edges but has the drawback of eliminating small details, due to the increased amount of blurring. Therefore, a neighborhood of 18 was found the best alternative in most cases. Summing up, for smoothing of brain MRIs anisotropic diffusion was perceived most useful. For segmenting the smoothed MRI, Tukey Biweight nonlinear diffusion was perceived best by the users because of its speed and its easy to tune parameters.

4 Discussion and Conclusion

This paper introduced an application which uses diffusion-based methods for smoothing and segmentation of volume data while providing real-time high-quality visualization. The fast and highly interactive diffusion implementation along with the real-time volume rendering offers increased flexibility and makes diffusion-based segmentation feasible for daily usage. The results show that diffusion is easier applicable for image smoothing than for segmentation purposes. Our interactive visualization, however, allows for fast intervention and stopping of the flow. Generally, the anisotropic diffusion turned out to be more edge enhancing than the isotropic diffusion, however it usually needs more iterations and is more time consuming. We embedded our approach into a surgical planning application for neurosurgeons. We were able to extract the brain and enhance the quality of DVR, thus enabling surgeons to get a better 3D impression of important structures. However, we still have to perform a clinical study to correctly evaluate our proposed application. In the future we want to incorporate more user interaction possibilities (e.g. to draw barriers which stop the diffusion process) and investigate in the automatic setting of diffusion parameters. Additionally we want to combine our method with other segmentation approaches (e.g. level sets) to further improve segmentation quality.

5 Acknowledgments

This research project was funded by the Austrian KPlus project and by AGFA Healthcare Vienna. The images are courtesy of the Neurosurgery Department at the Medical University of Vienna.

References

1. Weickert J. A Review of Nonlinear Diffusion Filtering. In: Lecture Notes in Computer Science; Proceedings of the First International Conference on Scale-Space Theory in Computer Vision; 1997. p. 3–28.
2. Suri JS, Wu D, Gao J, Singh S, Laxminarayan S. A Comparison of State-of-the-Art Diffusion Imaging Techniques for Smoothing Medical/Non-Medical Image Data. In: Proceedings of International Conference on Pattern Recognition (ICPR'02); 2002. p. 508–517.

3. Sherbondy A, Houston M, Napel S. Fast Volume Segmentation With Simultaneous Visualization Using Programmable Graphics Hardware. In: Proceedings of IEEE Visualization; 2003. p. 171–176.
4. Owens JD, Luebke D, Govindaraju N, Harris M, Krüger J, Lefohn AE, et al. A Survey of General-Purpose Computation on Graphics Hardware. In: Eurographics 2005, State of the Art Reports; 2005. p. 21–51.
5. Rumpf M, Strzodka R. Nonlinear Diffusion in Graphics Hardware. In: EG/IEEE TCVG Symposium on Visualization VisSym; 2001. p. 75–84.
6. Hopf M, Ertl T. Accelerating 3D convolution using graphics hardware. In: IEEE Visualization; 1999. p. 471–474.
7. Perona P, Malik J. Scale space and edge detection using anisotropic diffusion. *IEEE Transactions in Pattern Analysis and Machine Intelligence*. 1990;12(7):629–639.
8. Hadwiger M, Berger C, Hauser H. High-Quality Two-Level Volume Rendering of Segmented Data Sets on Consumer Graphics Hardware. In: Proceedings of IEEE Visualization; 2003. p. 301–308.
9. Ghita O, Robinson K, Lynch M, Whelan PF. MRI diffusion-based filtering: A note on performance characterisation. *Computerized Medical Imaging and Graphics*. 2005;29(4):267–277.
10. Levoy M. Display of Surfaces from Volume Data. *IEEE Computer Graphics and Applications*. 1988;8:29–37.
11. Rezk-Salama C, Engel K, Bauer M, Greiner G, Ertl T. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-Textures and Multi-Stage Rasterization. In: Proceedings of SIGGRAPH/Eurographics Workshop on Graphics Hardware; 2000. p. 109–118.
12. Krüger J, Westermann R. Acceleration Techniques for GPU-based Volume Rendering. In: Proceedings of IEEE Visualization; 2003. p. 287–292.
13. Black MJ, Sapiro G, Marimont D, Heeger D. Robust anisotropic diffusion. *IEEE Transactions on Image Processing*. 1998;7:421–432.
14. Scharlach H, Hadwiger M, Neubauer A, Wolfsberger S, Bühler K. Perspective Isosurface and Direct Volume Rendering for Virtual Endoscopy Applications. In: Eurovis/IEEE-VGTC Symposium on Visualization; 2006. p. 315–323.
15. Chu V, Hamarneh G. MATLAB-ITK Interface for Medical Image Filtering, Segmentation, and Registration. In: *Medical Imaging, Proc. of SPIE*. vol. 6144; 2006. .
16. Castellanos J, Rohr K, Tolxdorff T, Wagenknecht G. Automatic parameter optimization for denoising MR data. In: Proceedings of MICCAI; 2005. p. 320–327.

List of Figures

1	The segmentation processing pipeline.	4
2	Interactive visualization of the ongoing segmentation.	6
3	Left: Segmented and extracted brain. Left center: Two-level volume rendering showing the segmented brain and skull. Right center: Improved visualization of FCDs. Right: Segmentation error.	8