Interactive 3D Segmentation as an Example for Medical Visual Computing

Martin Urschler,
Alexander Bornik,
Eva Scheurer,
Thomas Pock,
Horst Bischof, Graz

Kurzfassung

Aufgrund der Vielfalt an potentiellen klinischen Anwendungsgebieten ist die Segmentierung medizinischer, volumetrischer Datensätze ein wichtiges Forschungsgebiet. Um klinische Relevanz und praktische Akzeptanz unter Radiologen und Ärzten zu erreichen, müssen generische, interaktive 3D Segmentierungsalgorithmen einfach zu bedienen und akkurat sein. Weiters ist ständiges und in Echtzeit dargestelltes Feedback für den Benutzer essenziell. In dieser Arbeit präsentieren wir einen neuartigen 3D-Segmentierungsansatz der mithilfe moderner 3D Grafikkarten Benutzerinteraktion, Segmentierung und volumetrische Visualisierung in ein gemeinsames Rahmenwerk integriert. Dies ist ein Beispiel für das Zusammenwirken von Computer Grafik und Computer Vision in einem Gebiet, das auch als Visual Computing bezeichnet wird. Direkte 2D und 3D Interaktion mit großen, volumetrischen Datensätzen wird mit einem Segmentierungsalgorithmus kombiniert, der als konvexes Energieminimierungsproblem definiert ist. Dieses global optimale Segmentierungsresultat und dessen Ausbreitung und Entwicklung während des Minimierungsprozesses wird kontinuierlich über eine hardware-beschleunigte Volume-Rendering-Engine visualisiert. Durch die integrierte Implementierung dieser Komponenten auf der Grafikkarte erhält man ein interaktives Echtzeit 3D Segmentierungssystem, welches Benutzerinteraktion auf das Nötigste reduziert. Die Einsatzfähigkeit des Systems zur Lösung praktischer Segmentierungsaufgaben wird anhand quantitativer und qualitativer Auswertungen gezeigt.

Schlüsselwörter: Interaktive 3D Segmentierung, Echtzeit Volumsvisualisierung, Volumens-Raycasting, Intuitive 3D Segmentierung

Abstract

Segmentation of medical volume data sets (i.e., partitioning images into a set of disjoint regions representing different semantic objects) is an important research topic due to its large number of potential clinical applications. In order to get accepted by physicians and radiologists a generic, interactive 3D segmentation algorithm has to be simple-to-use, accurate, and show immediate feedback to the user. In this work we present a novel 3D segmentation paradigm that effectively combines interaction, segmentation and volumetric visualization in a single framework integrated on a modern graphics processing unit (GPU). This is an example of the fruitful combination of computer graphics and computer vision, a field nowadays called visual computing. Direct interaction with a large volumetric data set using 2D and 3D painting elements is combined with a segmentation algorithm formulated as a convex energy minimization. This globally optimal segmentation result and its evolution over time is continuously visualized by means of a hardware accelerated volume rendering along with the original data. By implementing all of these components on a GPU, a highly responsive interactive 3D segmentation system requiring minimal user interaction is achieved. We demonstrate quantitative and qualitative results of our novel approach on liver and liver tumor segmentation data where a manual ground truth is available.

Keywords: interactive 3D segmentation, real-time volume rendering, volume raycasting, immediate feedback 3D segmentation

1. Introduction

Visual computing is a discipline that views computer graphics and computer vision, historically two separate disciplines, from a common perspective. The confluence of these two fields has led to many new insights and applications.Medical visual computing is a prototypical example that profits significantly from this convergence. In this paper we report on the recent work on an interactive segmentation system. The application requires computer graphic methods for 3D volume rendering and computer vision for segmentation. Since the system is designed to be used in an interactive manner, all these tasks have to be achieved at interactive framerates (10–15 fps). We accomplish this challenging problem by implementing all algorithms on modern graphics hardware. The re-

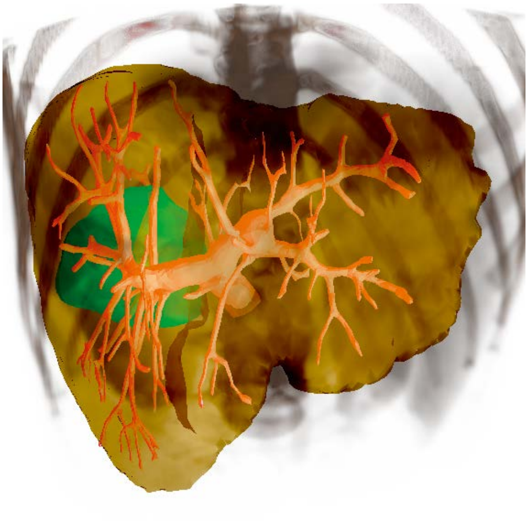

Fig. 1: Volume rendering of segmentation results from a liver CT data set. The segmented liver (yellow) is shown alongside the original volume data set, the extracted liver vessel tree (red) and a tumor (green).

sulting system is the first one that solves this complex task for realistic volume sizes. See Figure 1 for an example segmentation result.

Image segmentation is one of the key problems in computer vision with applications in a large number of domains. Medical image analysis is one such domain, where the delineation of different anatomical structures is a topic of significant research and clinical interest. Applications involving medical volume data sets exhibit a high degree of difficulty due to anatomical and pathological variations. Thus, fully automatic segmentation algorithms are seldom feasible in practice. Additionally, the routine way of diagnosis is centered around human operators. Thus, segmentation algorithms are required to assist physicians. As a consequence a lot of research in medical image segmentation is focusing on semi-automatic and interactive systems, where feedback is given to the physician during diagnostic investigations.

In interactive segmentation methods, users are required to highlight some regions as a prior, mostly drawing some kind of seeds or boundaries into an image. The LiveWire segmentation approach [1] is an early work for two-dimensional boundary delineation based on edge information. Later algorithms like GrabCut [2] combined color and edge information to separate image fore- and background using a Graph-Cut approach [3]. Very recently Sinop and Grady have proposed a related approach unifying graph cuts and the random walker algorithm for seeded image segmentation [4]. These works as well as our previous work in [5] were developed for 2D image segmentation applications. Some authors like Falcao and Bergo [6] or Poon *et al.*[7] have investigated an extension of the LiveWire methodology to 3D. Both approaches are restricted to boundary information and 3D user interaction is very hard. An example of recent work closely related to our method can be found in [8], who showed an interactive graph-cut based 3D segmentation approach called Live Surface. Their method uses innovative 3D interaction and provides immediate segmentation feedback. However, their approach loses flexibility due to their watershed-like pre-processing step. Errors at this stage cannot be corrected in the interactive setup. Moreover, their approach does not benefit from a parallel implementation, a possibility current consumer GPUs offer.

In this paper we show a novel, flexible 3D segmentation paradigm that integrates user interaction, segmentation and volumetric visualization into a single framework. This framework profits from the recent technological advances in processing power of modern GPUs and from their flexible programmability, e.g., using the CUDA framework [9]. This flexibility allows us to combine a segmentation algorithm based on a convex energy minimization formulation and a direct volume rendering (DVR) algorithm based on ray-casting into a single implementation on the GPU. User interaction is required to define constraints in the segmentation functional. These constraints lead to a globally optimal solution of the convex energy with respect to the user inputs which is calculated at interactive frame-rates. Immediate feedback of the current segmentation result is provided by combining the original volume and the segmentation in the volume rendering implemented on the same GPU. Thus, to our knowledge, we present the first interactive, state-of-the-art three-dimensional segmentation algorithm providing immediate feedback on the segmentation result that allows segmentation refinement and works on volume data sets of practical size.

2. System Description

In this section we develop our interactive, volume rendering based segmentation system. The framework implements 3D and 2D visualization views on 3D medical datasets and segmented structures, together with a segmentation algorithm including user interaction on the GPU. The software consists of three major components,

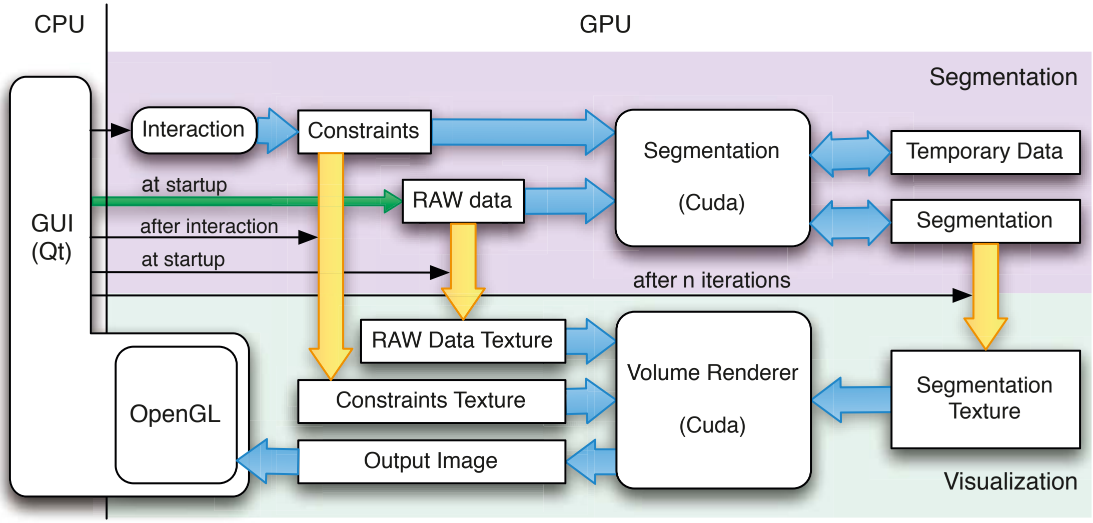

Fig 2: System Overview: The two main components segmentation and visualization run entirely on the GPU. Data transfers between the CPU based GUI are limited to system startup (file I/O), as indicated by the green arrow. Data access by the algorithmic components (rounded boxes) is shown using broad blue arrows, indicating the higher memory bandwidth. Inter-component data transfer triggered from the GUI (thin arrow) is indicated by yellow arrows.

the segmentation part, the visualization part and the GUI. An overview of the system is shown in Figure 2. Due to the GPU implementation, slow (2 GB/s) system bus data transfers occur only in the startup phase, when the 3D data is transferred to GPU memory. All further memory access happens on the GPU, where transactions are around 30 times faster than on the CPU. Even user interaction, like specification of foreground and background seed regions or segmentation refinement is performed on the GPU by triggering CUDA kernel functions, e.g., drawing a line from point A to B by painting into a 3D constraint data structure.

During the segmentation process the segmentation component runs continuously, reads data from the raw and the constraint dataset, as well as temporary and segmentation data. The latter two volumes are also modified during segmentation. The visualization component requires raw data, constraints and the segmentation as input and computes a 2D output image which is passed to the GUI component and visualized using OpenGL. Both components run independently but sequentially, since CUDA does not support concurrent kernel calls. Scheduling prioritizes rendering for the sake of interactivity.

All of the data used by visualization and algorithmic component is stored twice, which maximizes the efficiency of each algorithm since one can use optimal memory representation schemes for each task. However, memory redundancy requires copying, which is triggered by the controlling GUI component, predominantly after changes in the segmentation. Due to the very high GPU memory bandwidth copying is not a limiting factor in practice.

2.1 3D Volume Segmentation

For our 3D segmentation component we use an energy minimization model formulated in a variational framework. Given an input image in the domain $\Omega \subset \mathbb{R}^3$ we seek u, a binary labeling of the image into foreground $(u=1)$ and background $(u=0)$. Our minimization model is defined as

$$\min_u \left\{ \int_\Omega g \left| \nabla u(\boldsymbol{x}) \right| d\boldsymbol{x} + \lambda \int_\Omega u(\boldsymbol{x}) f(\boldsymbol{x})\, d\boldsymbol{x} \right\} \qquad (1)$$

The first term is a weighted total variation (TV) term penalizing discontinuities in the segmentation via the gradient of the labeling. The weighting g is related to edges in the input image $\mathrm{I}(\boldsymbol{x})$ by $g(\nabla \mathrm{I}(\boldsymbol{x})) = \exp\left(-\eta |\nabla \mathrm{I}|^\kappa\right)$ according to [10]. Here η and κ are chosen such that edges of interest from the gradient image $\nabla \mathrm{I}(\boldsymbol{x})$ are selected. Prior to edge calculation we use a ROF denoising scheme [11] implemented according to [12] to remove noise. With $u \in \{0,1\}$ the TV term is equivalent to the geodesic active contour (GAC) problem [13] minimizing the geodesic length of the segmentation interface with respect to the isotropic Riemannian metric g, a fact which was proven by Bresson et al. in [14]. A similar functional for segmentation has already been proposed in [15]. Note that minimizing the geodesic length can also be established by investigating the discrete binary labeling problem leading to Graph-Cut approaches [3]. The weight g restricts regularization of u to regions where the image gradient is low. The second term of (1) is a pointwise data-term inspired by Mumford-Shah-like energies [16], where a positive $f(\boldsymbol{x})$ forces $u(\boldsymbol{x})$ to be background, and a negative $f(\boldsymbol{x})$ forces $u(\boldsymbol{x})$ to be foreground.

Equation (1) is a convex energy formulation, however, our optimization domain, the binary labeling $u \in \{0,1\}$, is not a convex set. Thus, the optimization problem is not convex. A standard approach to achieve convexity is to relax u to the continuous domain $u \in [0,1]$. Since the TV energy satisfies the generalized co-area formula[17], thresholding the relaxed problem for any threshold $\mu \in (0,1)$ solves the original binary problem. The benefit of this convex formulation is that it allows to compute the binary labeling result u as the single global minimum of (1) after thresholding with any $\mu \in (0,1)$. We use $\mu = 0{,}5$ in our implementation. This way we arrive at a globally optimal segmentation algorithm w.r.t. the user constraints. We use the constraint term f of our convex functional to model different types of constraints on the labeling. Besides the case of $f = 0$ (which corresponds to a trivial global solution of the pure GAC energy) four different cases, which we use to drive our interactive segmentation, can be distinguished. With $f = -\infty$ we model a hard foreground constraint, where the algorithm forces $u = 1$. For a hard background constraint, we use $f = -\infty$, such that the algorithm forces $u = 0$. Using $f < 0$ models a weak foreground constraint, where the data term tries to make $u = 1$, however, depending on λ, the regularization can still force $u = 0$. A weak background constraint $f > 0$ works equivalently.

In our framework prototype, interactively specified weak constraints determine the gray value distribution of fore- and background objects, respectively. The details of calculating f make use of a simple histogram matching procedure, where we compare each voxel to the fore- and background distributions modeled as histograms. Here, of course different constraint terms are possible, e.g., texture features, similarity to prelearned gray value distributions or prior shape and appearance models. Hard constraints may be specified to remove or add certain parts of a segmentation result for interactive refinement.

Our GUI component provides the user with a 3D view and optional 2D views (axial, coronal or sagittal) on the data. Interaction, e.g., specification of the weak and hard constraints, selection of regions of interest, and segmentation refinement is possible using painting tools. Figure 3 shows 3D and 2D views with examples of constraint painting.

2.1.1 Numerical Implementation

To solve our energy minimization problem (1), we make use of a primal-dual algorithm inspired by [18] where a gradient descent on the primal and a gradient ascent on the dual variable are combined to find the global minimizer as a saddle point in the primal-dual formulation. The resulting numerical scheme can very efficiently be implemented in parallel on a GPU. We introduce a dual variable $\boldsymbol{p}$ defined as

$$g(\boldsymbol{x})|\nabla u| = \max_{||\boldsymbol{p}|| \leq g}\{\boldsymbol{p} \cdot \nabla u\}. \tag{2}$$

By reformulating Equation 1 using $\boldsymbol{p}$ we arrive at the primal-dual formulation of energy 1:

$$\min_{u} \max_{||\boldsymbol{p}|| \leq g}\left\{\int_{\Omega} \boldsymbol{p} \cdot \nabla u d\boldsymbol{x} + \lambda \int_{\Omega} u \cdot f d\boldsymbol{x}\right\}. \tag{3}$$

This is an optimization problem in two variables which has to be solved by alternately minimizing with respect to u and $\boldsymbol{p}$ until convergence. This leads to a projected gradient descent scheme, additional details about this optimization procedure and its parallelized GPU implementation can be found in [18].

1. **Primal update:** For the primal update we have to derive Equation 3 according to u:

$$\frac{\partial}{\partial u}\left\{-\int_{\Omega} u \nabla \cdot \boldsymbol{p} d\boldsymbol{x} + \lambda \int_{\Omega} u \cdot f d\boldsymbol{x}\right\} = -\nabla \cdot \boldsymbol{p} + \lambda f, \tag{4}$$

where we used the integral theorem, stating $\int \boldsymbol{p} \cdot \nabla u = -\int u \nabla \cdot \boldsymbol{p}$. Performing a gradient descent update scheme this leads to

$$u^{n+1} = \Pi_{[0,1]}\left(u^{n} - \tau_{P}\left(-\nabla \cdot \boldsymbol{p} + \lambda f\right)\right) \tag{5}$$

where the projection Π towards the binary set [0,1] can be done with a simple thresholding step, and τ_P denotes the timestep.

2. **Dual update:** Deriving Equation 3 according to $\boldsymbol{p}$ we get:

$$\frac{\partial}{\partial p}\left\{\int_{\Omega} \boldsymbol{p} \cdot \nabla u d\boldsymbol{x} + \lambda \int_{\Omega} u \cdot f d\boldsymbol{x}\right\} = \nabla u \tag{6}$$

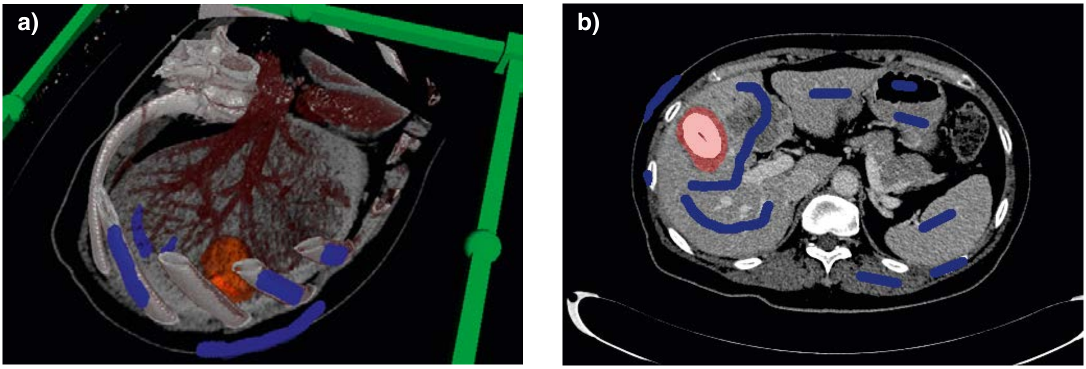

Fig. 3: User Interface Screenshots: (a) 3D DVR of the original dataset, isosurface rendered segmented liver tumor (brown) and segmentation constraints (blue) interactively painted on the DVR surface and the cutting plane. (b) 2D axial view of the same dataset with additional segmentation constraints and tumor.

with the additional constraint $||\boldsymbol{p}|| \le g$. This results into a gradient ascent method with a trailed reprojection to restrict the length of $\boldsymbol{p}$ to the weight g:

$$\boldsymbol{p}^{n+1} = \Pi_{B_0^g}\left(\boldsymbol{p}^n + \tau_D \nabla u\right) \tag{7}$$

Here B_0^g denotes a d-dimensional ball centered at the origin with the radius g. The reprojection onto B_0^g can be formulated with

$$\Pi_{B_0^g}\left(q\right) = \frac{q}{\max\left\{1, \frac{||\, q\, ||}{g}\right\}} \tag{8}$$

The result of this stage is a segmentation achieved after thresholding u. The numerical primal-dual scheme of the segmentation procedure can be easily parallelized and entirely implemented on the GPU using the CUDA programming language, which is our first component of the interactive segmentation algorithm.

2.2 Volume Rendering

To visualize medical 3D data direct volume rendering is a commonly used technique offering high image quality. We use volume ray-casting [19], an image-based technique, which can nowadays be implemented on the GPU. Unlike most existing GPU ray-casters we use a CUDA implementation, allowing greater flexibility on combination of data from different sources (i.e., raw data and segmentation) than previous shader based approaches like [20]. Our proposed ray-caster first renders bounding boxes of all regions obtained by splitting the raw dataset, segmentation and constraints into e.g., 32^3 voxel regions, where the application of transfer functions indicates impact on the visual output. Bounding box calculation also uses CUDA while OpenGL is only used in a rasterization stage to obtain two depth buffers from which ray entry and exit points can be calculated. In future implementations this rasterization of the bounding boxes will be done in CUDA, too, avoiding some copying overhead.

Since volume rendering is the process of visualizing data sampled from a scalar function, a ray traversal is started for each pixel and values are obtained by evaluating the transfer function of the raw datasets. The segmentation and the constraint datasets are visualized using iso-surface ray-casting, therefore, the results evaluating the raw dataset transfer function are intermixed with the iso-surface value. In addition, all different representations of the dataset can be clipped individually, using a cutting plane showing the underlying volume data. Clipping can also be done with a user-defined offset, which makes it easier to visually verify the segmentation progress and the final result. We use a stochastic sampling to avoid fringe artifacts.

3. Experimental Results

We implemented a first prototype of our proposed framework on the GPU using the CUDA programming language [9]. All results were calculated on a desktop PC (Intel 3.16 GHz Dual Core2-Duo with 3GB RAM) under 64-bit Linux. The PC contains two separate graphics units, a GeForce GTX 280 with 1 GB of graphics RAM used for visualization and a Tesla C1060 Computing Processor with 4GB of graphics RAM. The large amount of graphics RAM allows to work with data sets of practical size from 256^3 up to sizes of around 512×512×200 voxels. For these volume sizes we achieve an interactive framerate of around 15-20 frames per second without concurrent segmentation and 5-10 frames per second with activated segmentation. This depends, of course, on the window size of the rendering, the values are given for a resolution of 800×800 pixels

We performed experiments on two segmentation problems with annotated ground truth taken from the two recent MICCAI Segmentation Challenge workshops [21, 22]. In both experiments, segmentations were performed by a non-radiologist user experienced with medical image analysis. Our setup consisted of the segmentation framework and a 2 minute time-limit for segmentation per dataset. The user was allowed to draw weak and hard constraints and to perform segmentation refinement on the data.

In our first experiment we segmented the 20 liver datasets from the training data of the 2007 Grand Challenge. Since the ground truth segmentation for this data is readily available we compared our segmentation with it using the provided evaluation tool. Five measures are calculated by this tool, the volumetric overlap error (VOE), the relative absolute volume difference (RVD), the average symmetric surface distance (ASD), the root-mean-square surface distance (RMSSD), and the maximum symmetric surface distance (MSD)[21]. All of these values are zero for perfect agreement of segmentation and ground truth. Furthermore, a score for each measure is calculated by comparing it to a reference value of a human operator. The mean of these five scores is the total score of a data set, where 75 is a reference value for a human segmentation (see [21] for further details). In Table 1 we present our results compared to the reference values. These results indicate a reasonable performance, although we are not as good as the best approaches from the competition [21]. This is due to our tight time-limit and especially since we have not yet optimized our parameters of the segmentation framework to the task, but simply used our preliminary prototype with default settings.

Our second experiment consisted of 4 data sets containing a total of 10 liver tumors. This data from the 2008 Grand Challenge [22] was more difficult to segment since some of the tumors were very hard to distinguish from the liver tissue visually. The evaluation measures were the same as in the first experiment. The human observer reference score is 90 in this experiment, which is hard to achieve in practice as can be seen by the results of the workshop attendees ranging between 38 and 73 points. With our system we get a mean score of 50 points. Currently the main bottleneck in our approach is the histogram matching in the segmentation, which is not very well suited for this type of problem. Figure 4 shows a qualitative result from the liver segmentation.

To underline the generic applicability of our approach we now show some more qualitative segmentation results from different application areas. In Figure [1] a liver, vessel tree and liver tumor segmentation of an abdominal CT data set (256^3) is visualized as a DVR together with the original data. Figure 5 depicts a brain surface segmentation result from a CT data set together with the volume rendering focusing on the skull. Finally, we also produced a segmentation result on a very challenging segmentation task in the context of clinical forensic imaging. We used a T2-weighted MRI data set (512×512×30) of a human thigh showing a subcutaneous hematoma, with the segmentation result shown in red in Figure 6.

4. Conclusion and Future Work

In this paper we presented a first prototype of a segmentation framework, which combines a state-of-the-art segmentation algorithm and 3D visualization into a highly interactive framework. Even though neither the user interface, nor the algorithmic and the visualization parts have been optimized yet, the system already delivers reasonable results in terms of segmentation quality and interaction time. Nevertheless, future work will be directed towards optimization of the components. In the algorithmic part we intend to develop a new adaptive numerical solver, which should speed up the segmentation process by avoiding unne-

Experiment	VOE		RVD		ASD		RMSSD		MSD		Total
	[%]	Score	[%]	Score	[mm]	Score	[mm]	Score	[mm]	Score	Score
liver reference	6.4	75	4.7	75	1	75	1.8	75	19	75	75
liver	6.6	74	4.7	75	1	74	2.1	71	23	70	73
tumor reference	12.9	90	9.6	90	0.4	90	0.7	90	4.0	90	90
tumor	37	29	24.1	37	0.8	49	1.2	59	3.9	75	50

Tab. 1: Quantitative segmentation results of our first prototype on the liver and liver tumor grand challenge data sets. For details on the evaluation metrics refer to [21, 22].

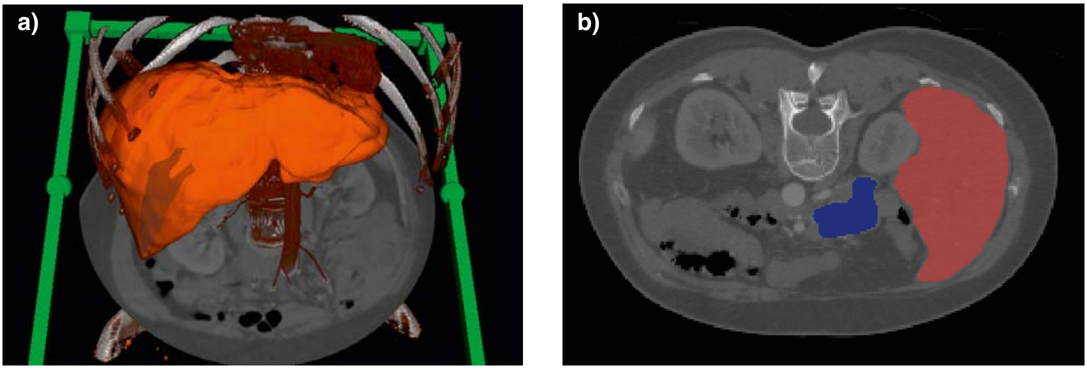

Fig. 4: *Liver segmentation example: (a) DVR emphasizing bones combined with isosurface rendering of the segmented liver and cutting plane showing original CT data. (b) 2D view of the same dataset with segmented liver in red and constraints in blue.*

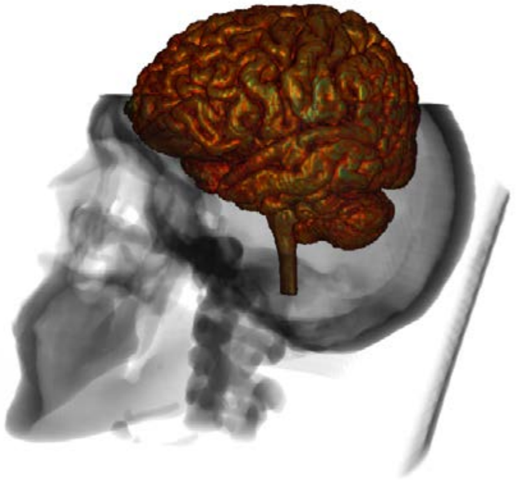

Fig. 5: *Segmentation result from a CT data set showing the brain surface together with the volume rendering focusing on the skull.*

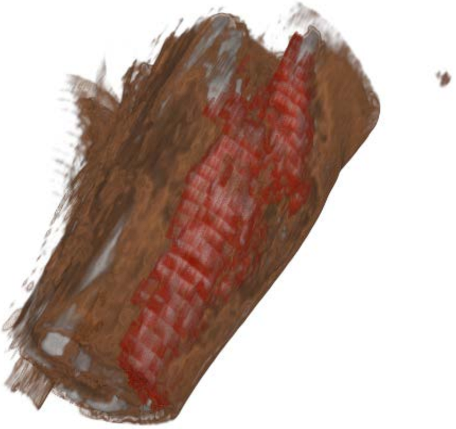

Fig. 6: *Segmentation result from an MRI data set of a human thigh containing a subcutaneous hematoma (red).*

cessary calculations in certain without losing the convexity property. Having more time for volume visualization the core framework could be integrated into a VR setup with true 3D user interaction, which would be helpful for segmentation refinement in particular. Furthermore, it would be interesting to incorporate prior shape knowledge into the whole system.

Literatur

[1] E. N. Mortensen and W. A. Barrett. Interactive Segmentation with Intelligent Scissors. *Graphical Models and Image Processing*, 60(5):349–384, Sep 1998.

[2] C. Rother, V. Kolmogorov, and A. Blake. “Grab-Cut”: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[3] Y. Y. Boykov and M.-P. Jolly. Interactive Organ Segmentation using Graph Cuts. In *MICCAI*, volume LNCS 1935, pages 276–286. Springer Verlag, 2000.

[4] A. K. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Proc. IEEE 11th International Conference on Computer Vision ICCV 2007*, pages 1–8, 14–21 Oct. 2007.

[5] M. Unger, T. Pock, W. Trobin, D. Cremers, and H. Bischof. Tvseg – interactive total variation based image segmentation. In *British Machine Vision Conference 2008*, Leeds, UK, September 2008.

[6] A. X. Falcao and F. P. G. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Transactions on Medical Imaging*, 23(9):1100–1108, Sept. 2004.

[7] M. Poon, G. Hamarneh, and R. Abugharbieh. Efficient interactive 3D Livewire segmentation of complex objects with arbitrary topology. *Computerized Medical Imaging and Graphics*, 32:639–650, 2008.

[8] C. J. Armstrong, B. L. Price, and W. A. Barrett. Interactive segmentation of image volumes with live surface. *Computers & Graphics*, 31:212–229, 2007.

[9] NVidia. NVidia CUDA Compute Unified Device Architecture programming guide 2.0. Technical report, NVIDA Corp., Santa Clara, CA, USA, 2008.

[10] J. Huang and D. Mumford. Statistics of natural images and models. *IEEE Conference on Computer Vision and Pattern Recognition*, 1:541–547, 1999.

[11] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D,* 60(1-4):259–268, 1992.

[12] A. Chambolle. An algorithm for total variation minimizations and applications. *Journal of Math. Imaging and Vision*, 20(1–2):89–97, 2004.

[13] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic Active Contours. *International Journal of Computer Vision*, 22(1):61–79, February 1997.

[14] X. Bresson, S. Esedoglu, P. Vandergheynst, J.-P. Thiran, and S. Osher. Fast global minimization of the active contour/snake model. In *Journal of Mathematical Imaging and Vision*, volume 28, pages 151–167, Norwell, MA, USA, 2007.

[15] S. Leung and S. Osher. Fast global minimization of the active contour model with TV-inpainting and twophase denoising. In *3rd IEEE Workshop Variational, Geometric, and Level Set Methods in Computer Vision*, pages 149–160, 2005.

[16] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and variational problems. *Communications on Pure and Applied Mathematics*, XLII(5):577–685, 1988.

[17] W. Fleming and R. Rishel. An integral formula for total gradient variation. *Arch. Math.*, 11:218–222, 1960.

[18] T. Pock, T. Schoenemann, G. Graber, H. Bischof, and D. Cremers. A convex formulation of continuous multilabel problems. In *Proc European Conference Computer Vision (ECCV 2008)*, volume 5304 of LNCS, pages 792–805, 2008.

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

[20] J. Beyer, M. Hadwiger, S. Wolfsberger, and K. Bühler. High-quality multimodal volume rendering for preoperative planning of neurosurgical interventions. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1696–1703, 2007.

[21] B. van Ginneken, T. Heimann, and M. Styner. 3D Segmentation in the Clinic: A Grand Challenge. In *Proceedings of the 3D Segmentation in the Clinic: A Grand Challenge Workshop of MICCAI 2007*, Brisbane, AU, 2007.

[22] X. Deng and G. Du. 3D Segmentation in the Clinic: A Grand Challenge II – Liver Tumor Segmentation. In *Proceedings of the 3D Segmentation in the Clinic: A Grand Challenge II Workshop of MICCAI 2008*, New York, US, 2008.

Acknowledgements

This work has been funded by the Ludwig-Boltzmann Institute for Clinical-Forensic Imaging, Graz, Austria. Dr. Pock was funded by the Austrian Science Fund (FWF) under the doctoral program Confluence of Vision and Graphics W1209. In addition, we want to thank DI Bernhard Kainz and DI Markus Unger from the Institute for Computer Graphics and Vision, Graz University of Technology for helping with the evaluations of the liver and liver tumor data.

Contact

Dr. Martin Urschler, Institut für Maschinelles Sehen u. Darstellen, TU Graz, Inffeldgasse 16, A-8010 Graz. Ludwig-Boltzmann Institut für Klinisch-Forensische Bildgebung, Universitätsplatz 4, A-8010 Graz.
E-mail: urschler@icg.tugraz.ac.at

Dr. Alexander Bornik, Institut für Maschinelles Sehen u. Darstellen, TU Graz, Inffeldgasse 16, A-8010 Graz. Ludwig-Boltzmann Institut für Klinisch-Forensische Bildgebung, Universitätsplatz 4, A-8010 Graz.
E-mail: bornik@icg.tugraz.ac.at

Dr. Eva Scheurer, Ludwig-Boltzmann Institut für Klinisch-Forensische Bildgebung, Universitätsplatz 4, A-8010 Graz.
E-mail: Eva.Scheurer@cfi.lbg.ac.at

Dr. Thomas Pock, Institut für Maschinelles Sehen u. Darstellen, TU Graz, Inffeldgasse 16, A-8010 Graz.
E-mail: pock@icg.tugraz.ac.at

Prof. Horst Bischof, Institut für Maschinelles Sehen u. Darstellen, TU Graz, Inffeldgasse 16, A-8010 Graz.
E-mail: bischof@icg.tugraz.ac.at

Vortragender

Univ.-Prof. Dr. Horst Bischof

1967	Saanen (Schweiz)
1985-1990	Studium der Informatik, TU Wien
1993	Dr. techn. TU-Wien
1993	Ruhr Universität Bochum
1998	Habilitation aus Angewandter Informatik, TU-Wien
2001-2003	Gastprofessor an der TU Graz
Seit 2004	Prof. für Computer Vision, TU Graz

Auszeichnungen:

12 Preise (Best Paper Awards) für wissenschaftliche Arbeiten, darunter Preis der Deutschen und Britischen Gesellschaft für Mustererkennung.

Insgesamt mehr als 400 begutachtete Veröffentlichungen.