

MITK-based segmentation of co-registered MRI for subject-related regional anaesthesia simulation

Christian Teich^a, Wei Liao^a, Sebastian Ullrich^b, Torsten Kuhlen^b,
Alexandre Ntouba^c, Rolf Rossaint^c, Marcus Ullisch^d and Thomas M. Deserno^a

^aDepartment of Medical Informatics, RWTH Aachen University,
Pauwelsstr. 30, 52074 Aachen, Germany;

^bVirtual Reality Group, Center for Computing and Communication,
RWTH Aachen University, Seffenter Weg 23, 52074 Aachen, Germany;

^cClinic of Anaesthesia, University Hospital Aachen, Pauwelsstr. 30, 52074 Aachen, Germany;

^dInstitute of Medicine, MR Group, Research Centre Jülich, 52425 Jülich, Germany

ABSTRACT

With a steadily increasing indication, regional anesthesia is still trained directly on the patient. To develop a virtual reality (VR)-based simulation, a patient model is needed containing several tissues, which have to be extracted from individual magnet resonance imaging (MRI) volume datasets. Due to the given modality and the different characteristics of the single tissues, an adequate segmentation can only be achieved by using a combination of segmentation algorithms. In this paper, we present a framework for creating an individual model from MRI scans of the patient. Our work splits in two parts. At first, an easy-to-use and extensible tool for handling the segmentation task on arbitrary datasets is provided. The key idea is to let the user create a segmentation for the given subject by running different processing steps in a purposive order and store them in a segmentation script for reuse on new datasets. For data handling and visualization, we utilize the Medical Imaging Interaction Toolkit (MITK), which is based on the Visualization Toolkit (VTK) and the Insight Segmentation and Registration Toolkit (ITK). The second part is to find suitable segmentation algorithms and respectively parameters for differentiating the tissues required by the RA simulation. For this purpose, a fuzzy c-means clustering algorithm combined with mathematical morphology operators and a geometric active contour-based approach is chosen. The segmentation process itself aims at operating with minimal user interaction, and the gained model fits the requirements of the simulation. First results are shown for both, male and female MRI of the pelvis.

Keywords: Segmentation, Simulation, Virtual Reality, Model Creation

1. INTRODUCTION

Regional anesthetics is a commonly used technique for blocking particular body regions (e.g. the extremities), without narcotising the patient all-over. Here the nerves, which innervate the specific region, are narcotised locally. For a coarse orientation, an anesthetist uses so-called outer leading structures (anatomical landmarks) to determine the position for puncture (Figure 1). After that, an electric stimulation-needle is used to locate the nerve cord within the tissue. An electric impulse triggers the contractions of associated muscles or a paresthesia stimulus. The strength is directly correlated with the distance between needle and the desired nerve. With decreasing distance, the potential of the electric impulse is being lowered. If a minimum distance has been found, a local anesthetics is released through the needle.¹ Typical scenarios for RA are operations on the upper extremities (e.g. brachial plexus block), lower extremities (femoral nerve block) and abdominal operations or cesarean sections (spinal block).

Corresponding author: Prof. Dr. Thomas M. Deserno, né Lehmann, Department of Medical Informatics, Aachen University of Technology (RWTH), Pauwelsstr. 30, D - 52057 Germany, email: deserno@ieee.org; RASim Project Group, email: contact@rasim.info, web: <http://www.rasim.info>

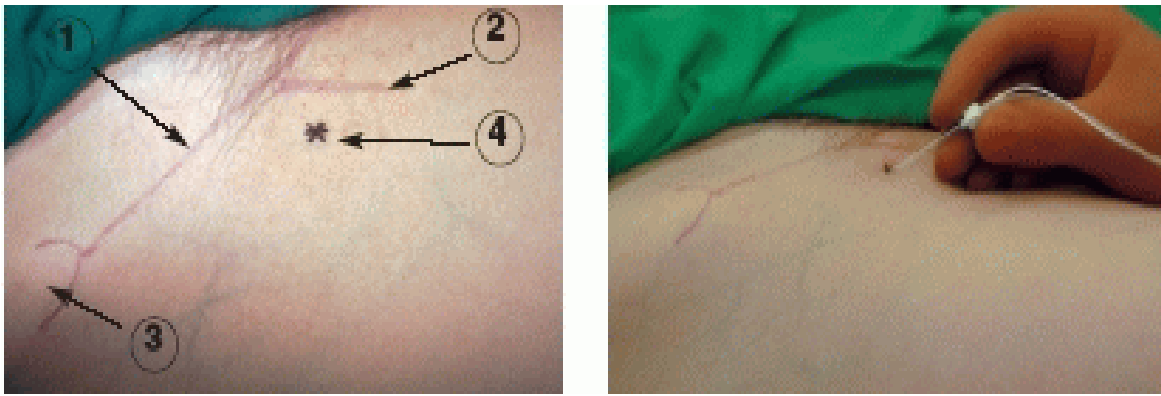


Figure 1. Femoral nerve block¹ : outer leading structures with 1) inguinal ligament 2) arteria femoralis 3) spina iliaca ant. sup. 4) puncture site (*left*); puncture with stimulation-needle (*right*)

To gain experience in regional anesthetics techniques, there is still no satisfying alternative to practicing in real situations. Beside the possibility to damage the nerve and adjacent structures like veins or even arteries with the needle itself, a physician can run the risk of an irreversible lesion while closing in on the nerve and trigger an electric impulse with a too large potential. The Regional Anesthetic Simulator (RASim)² project has been initialized in 2006 to develop a comprehensive virtual environment that allows to perform different block techniques on virtual patient models. A clinician can train all steps of a regional anesthesia procedure as described above and will get visual and haptic feedback of his actions. Beside the puncture procedure itself, the haptic response also includes the feel for the anatomic landmarks. Another feature is that the simulation is not restricted to a single dataset or a static model of one patient – the most important drawback of other approaches to this topic. So the underlying virtual patient database can cover a wide range of anatomical variances and even patient individual models.³ Our work covers the aspects of the patient-specific model creation and aims at developing a non-invasive MRI-based imaging protocol followed by a tissue differentiating segmentation that meets the requirements of the simulation. Focus is not set on the creation of a highly accurate and medical plausible model, like it is required by tools for diagnosing or operation planning. Instead, a scheme is developed for a semi-automatic segmentation and classification of the different tissues out of given 3D MRI data. The relevant tissue classes for the simulation are:

1. *bones*, which are needed for the skeleton model (animation) of the virtual patient and also as anatomical landmarks for the detection and location of the puncture site;
2. *musculature*, which is recommended for the simulation of contractions, while stimulating with an electric impulse. In some body regions, a clinician also has to penetrate musculature to address the specific nerve;
3. *blood vessels*, which are needed to detect the puncture site (pulsation inside a vessel). There will be no distinction between the arterial and the venous system;
4. *soft tissue* (including skin, subcutaneous fat and connective tissue) is needed for realistic representation of the model surface and to determine the resistance, while actually performing the puncture. Furthermore, the nerve and vessels are typically embedded in a surrounding protective coat of soft tissue; and
5. *nerve cords*, which are the base tissue for the simulation.

In this paper, we present a flexible and extensible framework that allows a user to perform the model creation for a broad variety of MRI data, representing a body region, where a blockade technique will be applied. Results are exemplarily shown for the lower extremities, in particular the pelvis.

2. METHODS

2.1 Data Acquisition

To collect suitable/feasible data for the project, there were some restrictions for the imaging modality to consider. The imaging process should get along without additional stress and must not be invasive for the patient. That excludes the use of computed tomography (CT) imaging and the injection of any contrast agents; in addition the acquisition time should be as short as possible. Therefore, contrast of MRI was improved by appropriately setting parameters and time constants involved in the relaxation processes of the tissue nuclei.

We initially tested several MRI sequences to enhance the nerve fibers itself for segmentation, but with less convincing results. Due to their small structure and the properties to share the same intensity band like the surrounding connective tissue, in which the nerves are embedded, current MRI technology cannot sufficiently display the nerves. However, since peripheral nerves follow the paths of blood vessels, they can be easily modeled along the vessel tree. A flexible system has been developed for interactively modeling and simulating peripheral nerve cords that allows to attached/connected control points to other (segmented) anatomical structures for realization of movement and functional dependencies.⁴ After all, two MRI sequences were chosen to gain sufficient information for a classification of the desired tissues:

1. a magnetic resonance angiography (MRA) without a contrast agent is used to extract the vascular system;
2. a T1-weighted MRI morphology sequence is applied to provide information on other tissues classes.

The sequences are produced with a Philips Achieva MR Scanner with a field strength of 1.5 Tesla. In both sequences, a 4-channel SENSE Body coil is applied to increase the signal-to-noise (SNR) ratio on the one hand and to shorten the scan times on the other hand. The MRA is a 2D time of flight (TOF) sequence with 128 slices and a dimension of 512 x 512 pixels; parameters: repetition time = 23.0 ms, echo time = 6.9 ms and flip angle = 23°. The morphologic MRI dataset is a 2D T1-weighted fast-field-echo (FFE) sequence with 128 slices and a resolution of 480 x 480 pixel; parameters: repetition time = 218.4 ms, echo time = 4.6 ms and flip angle = 80°. Both sequences are applied one after another and cover the same volume (i.e. both use the same field of view) without reposition the patient. So they are a priori co-registered and no additional registration of both datasets is needed. The overall time for data acquisition is currently about 10 minutes.

2.2 Implementation

For the implementation of our framework, the Medical Imaging Interaction Toolkit (MITK)⁵ is used, because it offers some interesting concepts matching our requirements. MITK is developed at the German Cancer Research Center (DKFZ) and combines/extends the Insight Toolkit (ITK)⁶ and the Visualization Toolkit (VTK).⁷ ITK offers a comprehensive library of algorithms for image processing, segmentation and registration tasks, but does not support interaction and visualization of image data. VTK instead is designed to visualize a wide range of scientific data in many different ways. It provides graphical user interface (GUI) support in form of render widgets and low-level support for interaction and manipulating the data such as panning, rotating, moving and scaling.

For medical image processing and visualization, it is straightforward to use a combination of both toolkits. One big drawback of this solution is the effort to keep data and the view(s) on the data consistent. On the one hand changes in the data to be visualized have to be propagated to the view(s) itself. On the other hand, if you are using multiple different views on the same data, changes in one view may have impact to the other displays, so they also have to be updated. MITK overcomes this gap by managing different data (input, intermediate/final processing results or geometric objects) in a so-called data tree, which represents logical dependencies and also defines the content of the views comparable to a scene graph.⁸ Furthermore we make use of the undo/redo concept in MITK, which is well suited for the creation of interactive applications. For building the GUI, the Qt library is used, because it is well supported by the MITK, respectively its Qt dependent part QMITK. For realization of the used filters and segmentation algorithms, we utilize the ITK and also VTK, as far as possible. Figure 2 shows the relations between the framework and the utilized toolkits and libraries.

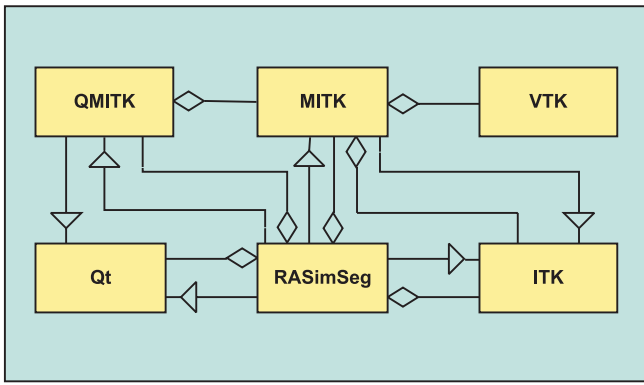


Figure 2. Relations between the RASimSeg framework and the utilized toolkits and libraries

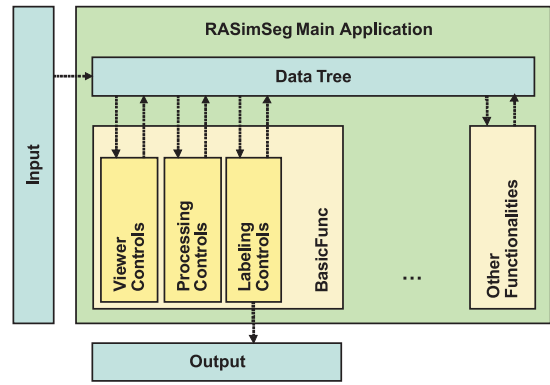


Figure 3. Application logic with its extensible modular structure

2.3 Framework and Graphical User Interface

MITK comes along with a basic application template that can be used to integrate so-called functionalities. This template serves as base of the RASimSeg application. The concept of the functionalities has been maintained to keep the application flexible in terms of adding additional functionalities in the future. The application logic is depicted in Figure 3. The communication between different parts (i.e. single functionalities) of the application is exclusively done over the common data tree. That is, data is changed or added to the tree and the change is announced by invoking signals on the node or the data tree itself.

The actual functionality of the RASimSeg application is implemented in *BasicFunc*. It consists of three parts: (i) the viewer controls, (ii) the processing controls, and (iii) the labeling controls. Basically, all of them can be regarded as self-contained functionalities. The main application layout is separated in two areas, a control area that holds the graphical user interface (GUI) of the currently selected functionality and the viewer area. Last provides an intuitive visualization of all inputs, intermediate and final results using the common view for volume data consisting of three orthogonal 2D-views and a 3D-widget for volume and surface rendering.

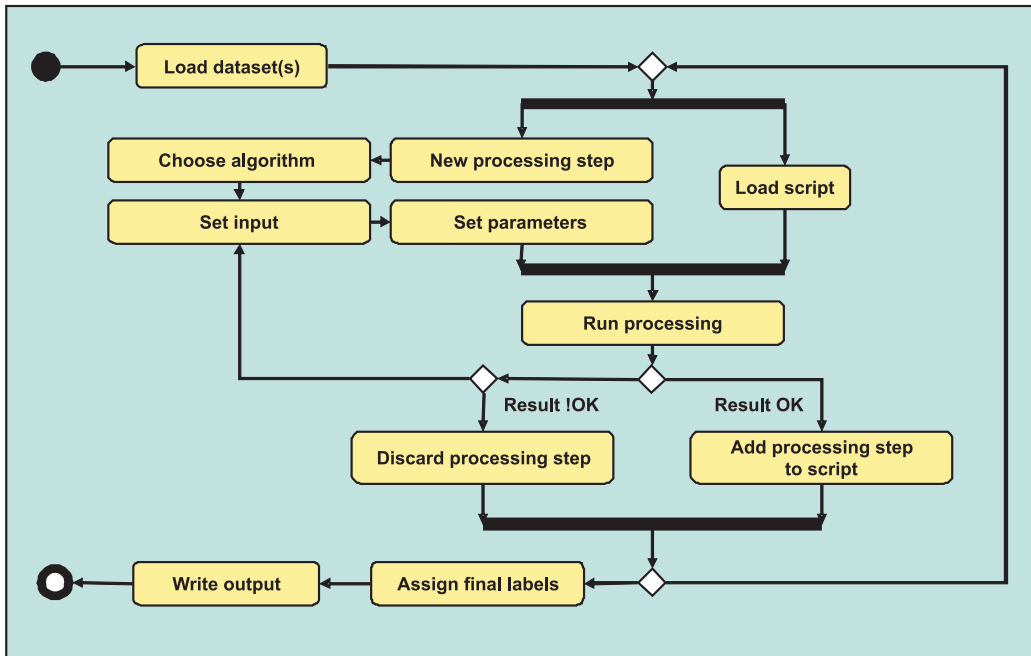


Figure 4. Activity diagram for the segmentation process

The framework allows to open and manage multiple input image data, not restricted to a specific imaging modality. Currently, we assume the datasets to be co-registered. However, support for 3D volume registration can be added easily if required.

In order to create a segmentation, the user can run successive processing steps by choosing an appropriate algorithm, setting the input and the required parameters. The algorithms are implemented as shared libraries and are loaded dynamically into the application via a plugin mechanism. After executing a single step or the processing pipeline, the result(s) can be inspected with the integrated viewer. At each state, the user can correct the outcome by setting more appropriate parameters and rerun the step(s) or even discard the result of the segmentation (Figure 4). Datasets loaded in the application share a common label (binary images) list identified by the step they are produced in and respectively their position in the data tree. There are three types of labels:

1. labels created as output of a single segmentation step,
2. user defined labels (for example seed points or a region of interest), and
3. labels constructed as logical operation on other labels.

These labels can be used either as mask or as input for a subsequent processing step, depending on the algorithm. After a segmentation session, the results of single steps can be combined to a final result dataset by assigning/mapping them to final labels representing the desired tissues by a unique iso-value and a color, respectively. The order of the processing steps including their inputs and parameters can be exported to an Extensible Markup Language (XML) file. A stored segmentation script that has been approved for a specific task can be imported and applied to another corresponding dataset. Before running a loaded script, the user has

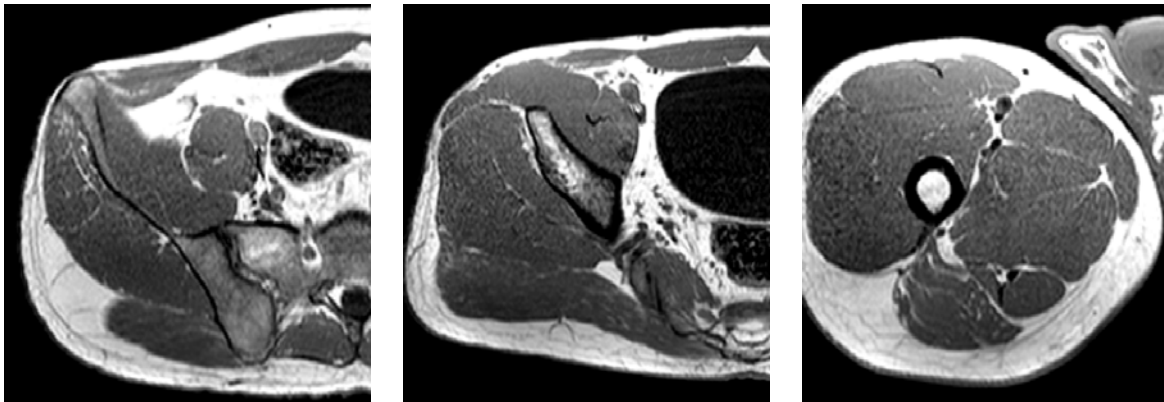


Figure 5. Sample slices from T1-weighted MRI showing variety of bones structures: parts of the pelvis (*left, middle*); middle part of the femur with broad (black) cortical bone (*right*)

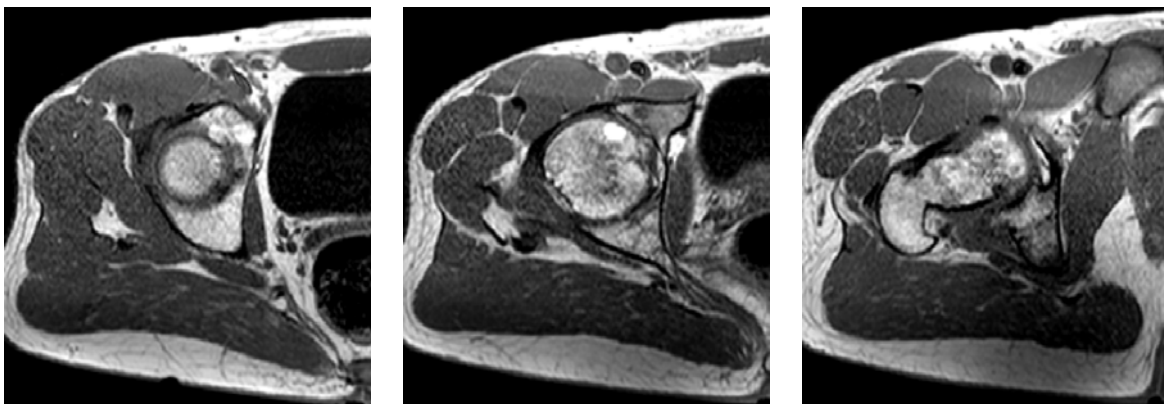


Figure 6. Sample slices from T1-weighted MRI showing the complexity of the femur joint (from superior to inferior)

to provide those parameters that could not be transferred between different patient data like user-specified seed points. He further has the possibility to change and adjust parameters of each step to optimize the segmentation result for the current dataset.

2.4 Segmentation

Usually, a special imaging technique is chosen for a tissue class to enhance and optimize the contrast for visualization and segmentation. So, suitable filter and segmentation algorithms had to be found that fit the MRI-data. In order to rerun a once created script on new datasets, we want the segmentation process to be preferably initialized automatically and run with minimal user interaction. We propose the following segmentation scheme for differentiating the desired tissue classes out of MRI data from the pelvis.

To extract the vascular tree from the 3D MRA, we first pre-process the data with an edge preserving smoothing

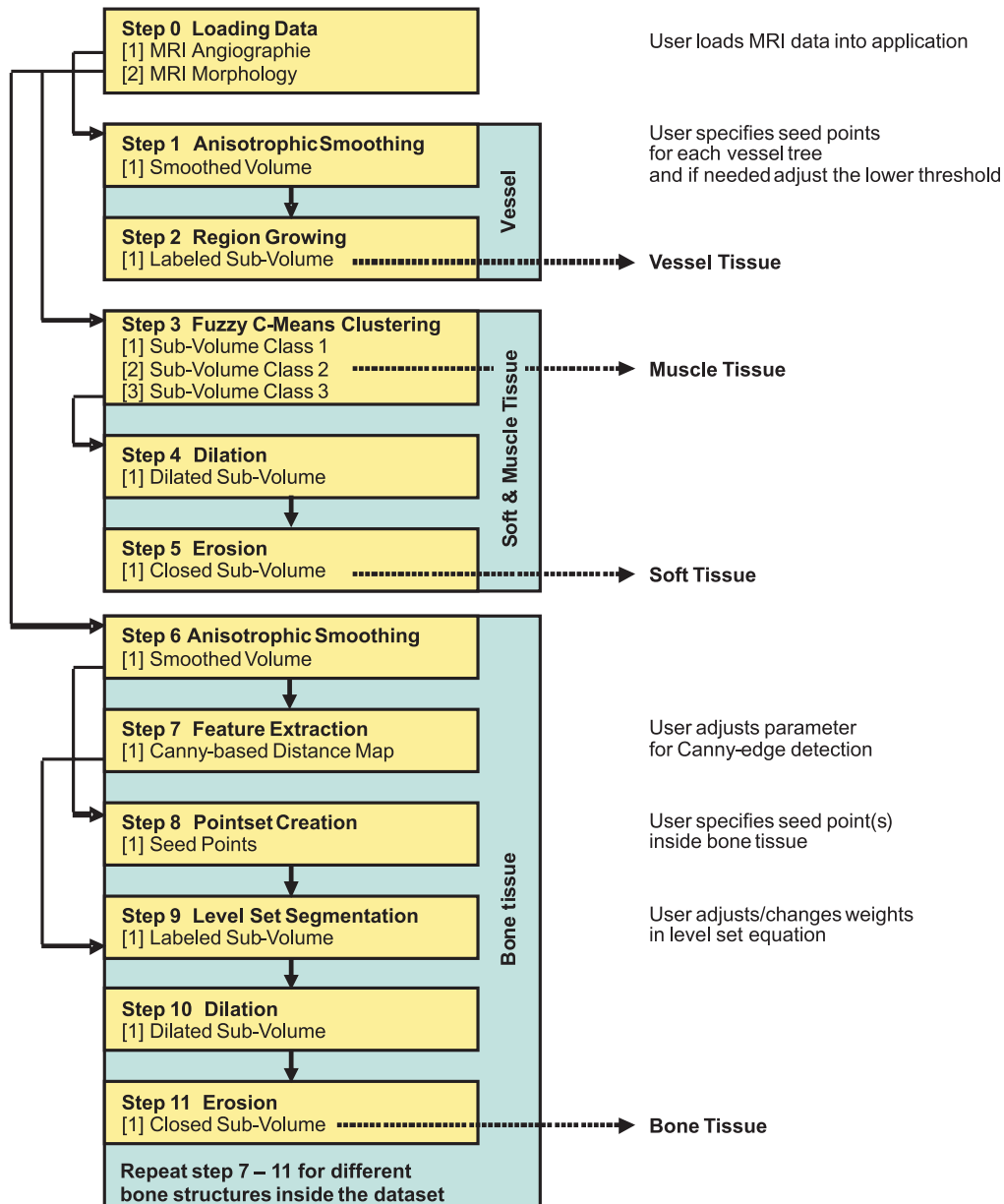


Figure 7. Proposed segmentation script

algorithm to reduce noise and local inhomogeneities and then apply a region growing algorithm on user provided seed points. Here, at least two seed points have to be specified, one for each vascular tree in the body region. The algorithm only considers the intensity values and simply includes adjacent voxels that lies within a fixed interval.

For the segmentation of the soft tissue, we follow the approach described by Positano et al.⁹ using a fuzzy c-mean clustering algorithm. We use an implementation based on the original proposed algorithm by Bezdek.¹⁰ The clustering algorithm is automatically initialized with a predefined number of three classes to separate the T1-weighted data into (i) background, (ii) soft tissue and (iii) a class containing the remaining tissue. Class 3 can be used for an appropriate differentiation of muscle tissue. The output of the clustering algorithm is sorted by the values of the cluster center (i.e. the intensity mean of each cluster), so there is a well-defined order of the produced labels for the further use in the processing pipeline. Mathematical morphology operators are applied to the soft tissue labels to smooth the result, particularly to smooth the body surface.

The segmentation of bone structures in MRI is a difficult and challenging task, especially in the region of the pelvis with its complex shape and the femoral joint. The major problem is that bones are composed of different substructures. There are cortical bone, trabecular bone with different levels of calcifications, and marrow cavities. Each of these structures has an individual MRI characteristic (i.e. intensity distribution), which makes it difficult to frame some common segmentation criteria based on intensity or texture features (Figure 5). In fact, the result of the clustering algorithm shows that bone tissue is unequally distributed to all classes. Furthermore, edge features derived from the gradient to the surrounding tissue are difficult to define. In particular, at the femur joint with the epiphysis and the cartilage interface (Figure 6), the intensity gradient at the boundary is weak compared to other regions of bones that have a cortical structure.

Nevertheless, we try to address the problem of the bone segmentation with a geometric deformable model approach, first described by Caselles et al.¹¹ and Malladi, Sethian & Vemuri.¹² Thereby, a contour or surface is initialized and is then evolved based on edge features until it fits the form of the bone structure. For the evolution of the interface, we utilize the generic level set framework of the ITK.¹³ Canny-edge detection is applied to emphasize the edges of bone boundaries. A distance map to these edges drives the level set evolution. The aspired aim to achieve a rough prior model for the level set segmentation with a hierarchical fuzzy c-means

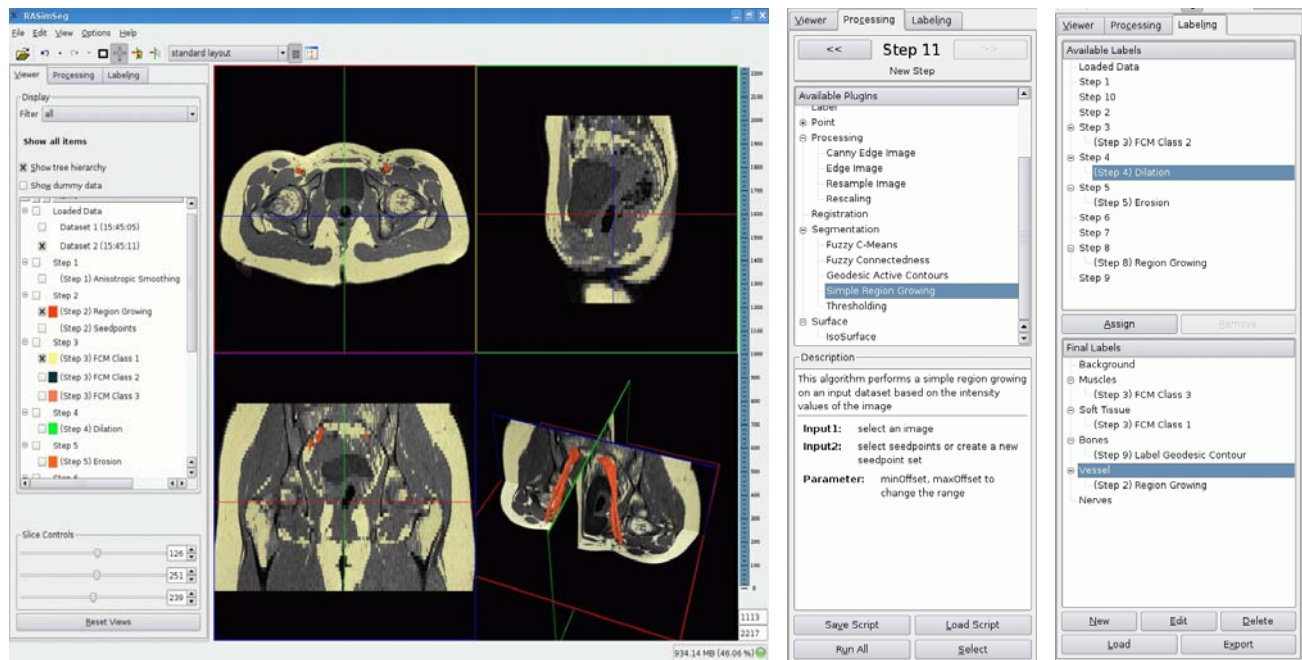


Figure 8. RASimSeg Application: Interactive viewer and controls (*left*); GUI used to control the image processing (*middle*); GUI for performing the final labeling (*right*)

clustering approach has not been promising, due to the non-uniform intensity distribution. For now, an initial model is provided by a region growing approach or simply by user-defined seed points. The drawback of this initialization is that each bone segment in the dataset has to be considered on its own. The proposed script is visualized in Figure 7.

3. RESULTS

Our framework is used to handle the segmentation of relevant tissues for RA-simulation from two given MRI data, an MR angiography and a T1-weighted morphology. In Figure 8, the GUI of the application is presented. The presented script, consisting of single image processing and segmentation steps, is used to create a virtual model of the patient. The script has been tested on two datasets. Figure 9 shows the results of the region growing approach for the vessel detection and the fuzzy c-mean clustering for differentiating soft and muscle tissue. To process all data the script takes about approx. 25 min on a Intel Dual Core 1.80Hz machine with 2 GB RAM for executing steps 1 to 5. The results show that the segmentation we achieve with the proposed scheme is suitable for the requirements in our context. Furthermore, they are reproducible and the only user input required is the specification of the initial seed points for the vessel extraction.

Figure 10 shows the results of differentiating the femoral bone in the dataset. Left and right bone are segmented in one run of the algorithm, using a single seed point in each structure. The major problem is that parameters used for the Canny-edge detection and in the level set algorithm have to be individually adjusted for different bone structures and different datasets.

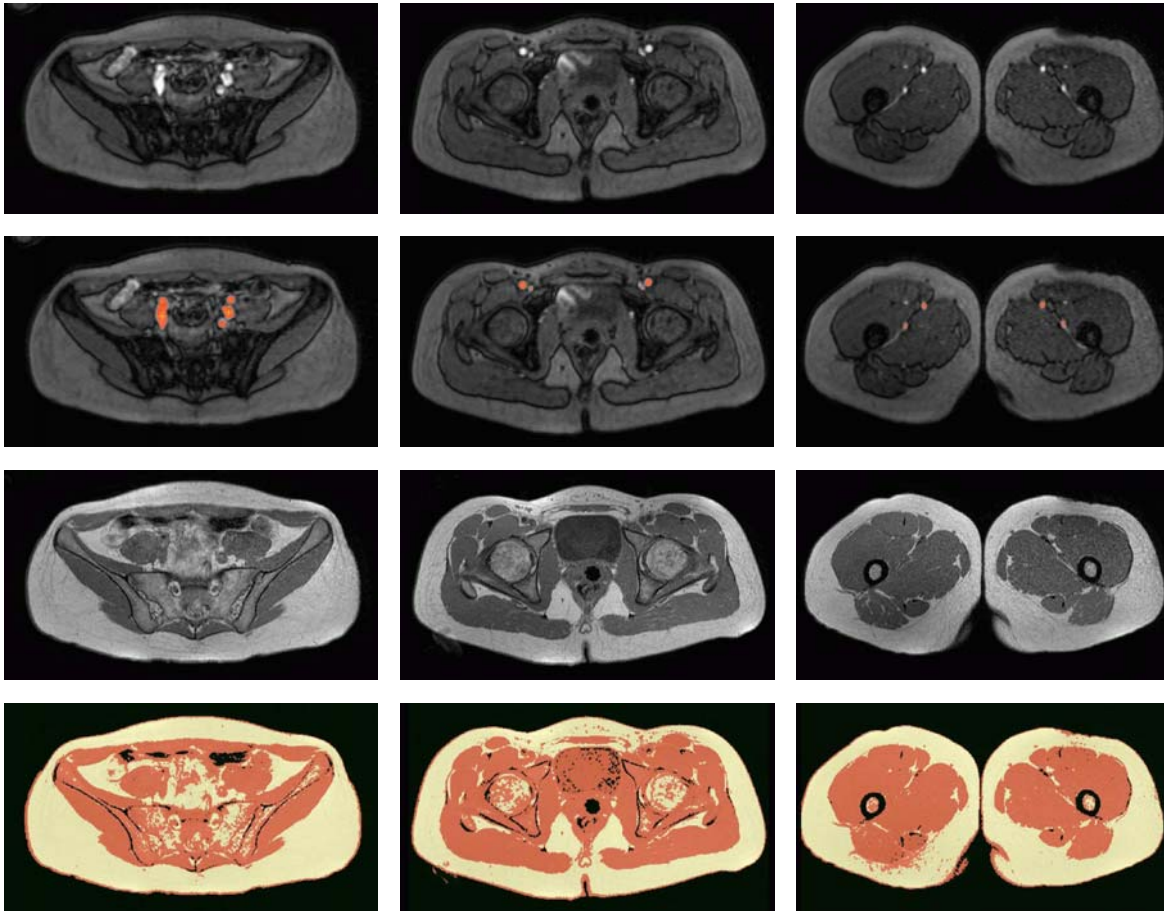


Figure 9. Results: Sample slices from the MRA (*top row*); results for the region growing initialized with a single point in each vessel tree (*2nd row*); corresponding slices from the morphologic MRI (*3rd row*); result of the clustering into three classes: background (black), soft tissue (yellow) and muscle (red) (*bottom row*)

In Figure 11, geometric representations of the segmented tissues are shown for two different MRI scans. For a better visualization, the iso-surface of the muscle tissue is hidden. The soft tissue is only visualized by the outer hull, representing the body surface. The segmentations have been achieved by running the proposed script on each MRI data, without further adjusting the algorithm parameters.

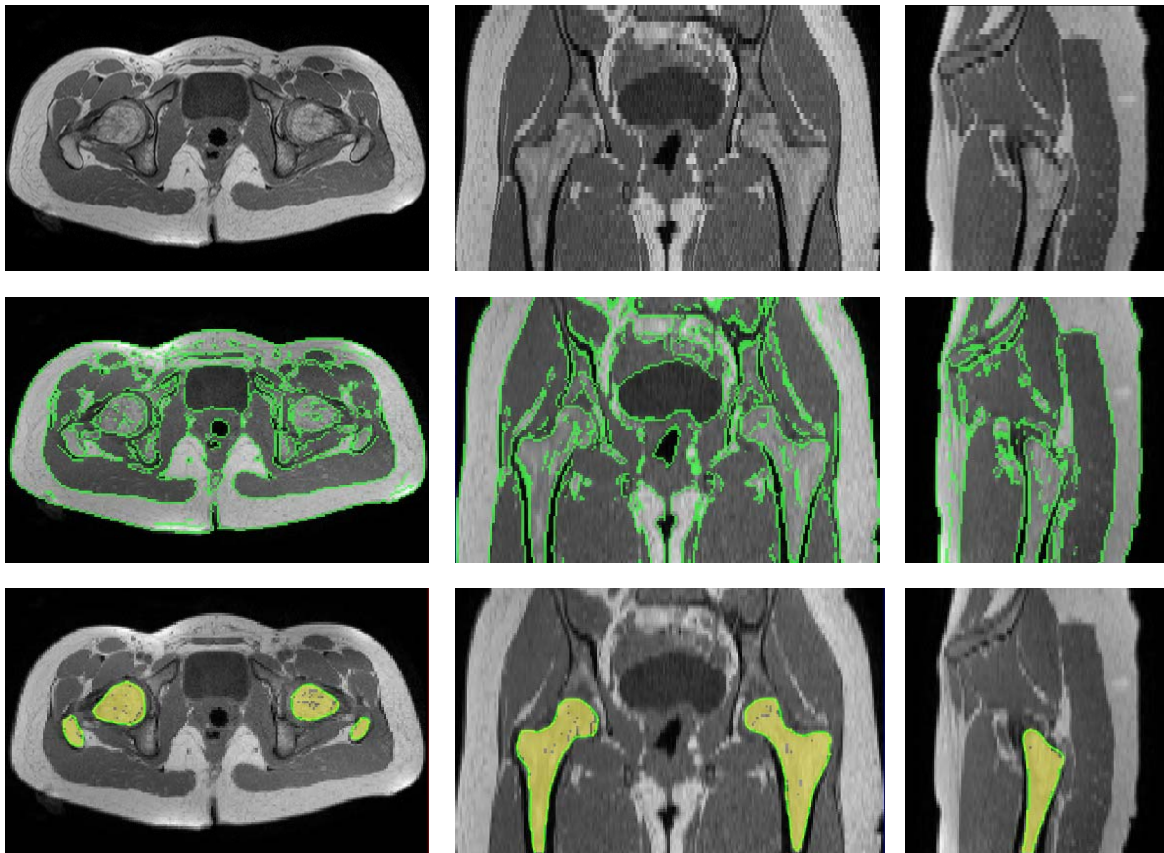


Figure 10. Results: Transveral, coronar and sagital view of the femoral bone (*top row*); result of a 3D Canny edge detection applied to the anisotropic smoothed image (*middle row*); result of the level set segmentation (yellow) and smoothed iso-surface extracted after a closing operation (green) (*bottom row*)

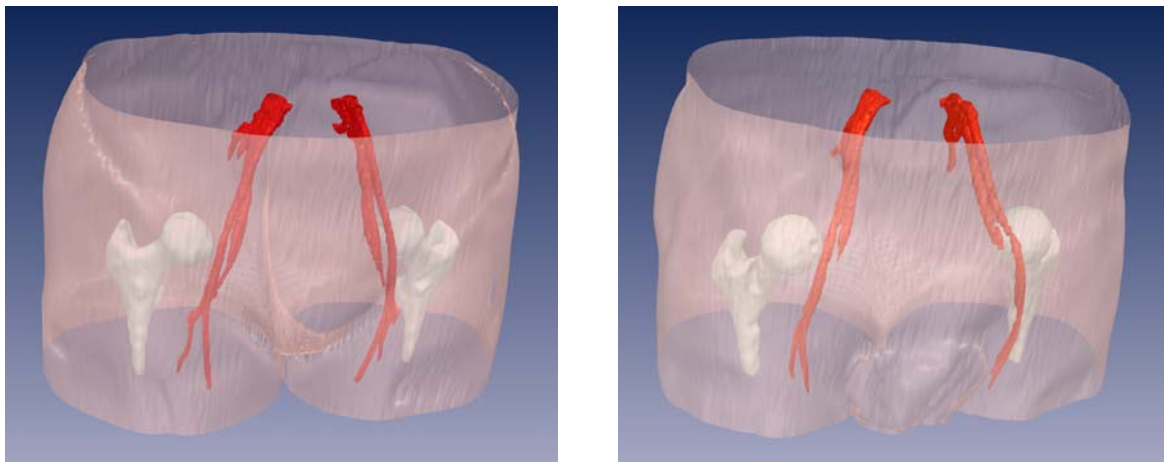


Figure 11. Iso-surfaces for skin, vessels and (incomplete) bone tissue extracted from two MRI scans using the proposed segmentation script without additional adjusting parameters: female (*left*); male (*right*)

4. DISCUSSION AND CONCLUSION

Segmentation of medical datasets is a non-trivial task. We provide a framework to establish subject-related data for VR-based training of regional anesthesia. The framework is capable to handle multiple input data and is designed to create a segmentation step-by-step. For now only, a few pre-selected image processing algorithms are integrated in the framework, but it is hold to be flexible and extensible. Using MITK, an easy-to-use and intuitive GUI is created to generate processing scripts for a specific task, input data, and body region. These scripts can be stored and applied to further datasets of corresponding tasks. The usability of the framework was proven exemplarily when creating a segmentation script for multi-modal MRI data of the pelvis (morphology and co-registered angiography). The simulation is not restricted to a single static model, but can cover a wide range of different patient types with anatomical variations for training or even models for a patient-individual preparation.³

Future work includes optimization of the segmentation results. In particular, the segmentation of the bone tissue can be improved with respect to the achieved results and the desired automation. Further evaluation of the segmentation results and of the re-usability of generated scripts based on a larger database will be performed.

ACKNOWLEDGMENTS

We would like to thank Prof. Dr. N. J. Shah from the Institute of Medicine (MR Group), Research Centre Jülich for fruitful discussions concerning the initial problems of the data acquisition. Furthermore, we thank Dr. C. Hohl and A. Buhl at the Department of Diagnostic Radiology, University Hospital Aachen, for the support on the data acquisition itself. The RASim project was funded by the START-Program (UK Aachen) and is partly supported by the German Research Foundation (DFG), grants no. KU 1132, LE 1108, RO 2000.

REFERENCES

- [1] Mehrkens H-H, Geiger PM. Tutorial peripheral regional anaesthesia. Ulm; 2002 [updated 2005 Sep; cited 2008 Jan]. Available from: <http://www.nerveblocks.net>.
- [2] RASim project [homepage on the Internet]. RWTH Aachen; 2006 [updated 2008 Jan]. <http://www.rasim.info>
- [3] Ullrich S, Fischer B, Ntoubas A, Valvoda JT, Prescher A, Kuhlen T, et al. Subject-based regional anaesthesia simulator combining image processing and virtual reality. In: Horsch A, Deserno TM, Handels H, Meinzer HP, Tolxdorff T, editors. *Bildverarbeitung für die Medizin 2007*; 2007. p. 166–71.
- [4] Ullrich S, Frommen T, Eckert J, Schütz A, Liao W, Deserno TM, et al. Interactive modeling and simulation of peripheral nerve cords in virtual environments. *Proc SPIE 2008*; 6918 in press
- [5] MITK – Medical imaging interaction toolkit [Online]. Medizinische und Biologische Informatik, DKFZ; [updated 2007 Dec; cited 2008 Jan]. Available from: <http://www.mitk.org>.
- [6] ITK – Insight Segmentation and Registration Toolkit [Online]; 1999 [updated 2007 Apr; cited 2007 Sep]. Available from: <http://www.itk.org>.
- [7] VTK – The Visualization Toolkit [Online]; [updated 2007 Jun; cited 2007 Sep]. Available from: <http://www.vtk.org>.
- [8] Wolf I, Vetter M, Wegner I, Nolden M, Böttger T, Hastenteufel M, et al. The medical imaging interaction Toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK. In: Galloway RL, editor. *SPIE Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display*. vol. 5367; 2004. p. 16–27.
- [9] Positano V, Gastaldelli A, Sironi AM, Santarelli MF, Lombardi M, Landini L. An accurate and robust method for unsupervised assessment of abdominal fat by MRI. *Journal of Magnetic Resonance Imaging*. 2004;20(4):684–9.
- [10] Bezdek JC. *Pattern recognition with fuzzy objective function algorithms*. New York: Plenum Press; 1981.
- [11] Caselles V, Catta F, Coll T, Dibos F. A geometric model for active contours. In: *Numerische Mathematik*. vol. 66; 1993. p. 1–31.
- [12] Malladi R, Sethian JA, Vemuri BC. Shape modeling with front propagation: a level set approach. *Trans Patt Anal Mach Intell*. 1995;17(2):158–75.
- [13] The ITK software guide [Online]. Ibáñez L, Schroeder W, Ng L, Ng CJ, the Insight Software Consortium; [updated 2005 Nov; cited 2007 Apr]. Available from: <http://www.itk.org/ItkSoftwareGuide.pdf>.