

Interactive Modeling and Simulation of Peripheral Nerve Cords in Virtual Environments

Sebastian Ullrich^a, Thorsten Frommen^a, Jan Eckert^a, Astrid Schütz^a,
Wei Liao^b, Thomas M. Deserno^b, Alexandre Ntouba^c,
Rolf Rossaint^c, Andreas Prescher^d and Torsten Kuhlen^a

^aVR Group, RWTH Aachen University, Seffenter Weg 23,

^bDepartment of Medical Informatics, RWTH Aachen University, Pauwelsstr. 30,

^cClinic for Anesthesiology, University Hospital Aachen, Pauwelsstr. 30,

^dInstitute of Neuroanatomy, University Hospital Aachen, Pauwelsstr. 30,
52074 Aachen, Germany

ABSTRACT

This paper contributes to modeling, simulation and visualization of peripheral nerve cords. Until now, only sparse datasets of nerve cords can be found. In addition, this data has not yet been used in simulators, because it is only static. To build up a more flexible anatomical structure of peripheral nerve cords, we propose a hierarchical tree data structure where each node represents a nerve branch. The shape of the nerve segments itself is approximated by spline curves. Interactive modeling allows for the creation and editing of control points which are used for branching nerve sections, calculating spline curves and editing spline representations via cross sections. Furthermore, the control points can be attached to different anatomic structures. Through this approach, nerve cords deform in accordance to the movement of the connected structures, e.g., muscles or bones. As a result, we have developed an intuitive modeling system that runs on desktop computers and in immersive environments. It allows anatomical experts to create movable peripheral nerve cords for articulated virtual humanoids. Direct feedback of changes induced by movement or deformation is achieved by visualization in real-time. The techniques and the resulting data are already used for medical simulators.

Keywords: Modeling, Surgical Simulation, Visualization

1. PURPOSE

There are only few datasets of peripheral nerves. Furthermore, most of these models are static and can only be found in proprietary formats within multimedia anatomy atlases. Two notable examples are: Interactive Complete Human Anatomy Series¹ (pre-rendered three-dimensional images, no export functionality) and VOX-ELMAN² (real three-dimensional volume rendering and mixed visualizations, based/focused upon Visible Human dataset). One reason for this lack of data is the difficulty in detecting nerves in medical imaging (such as CT and MRI). In spite of that, it is still possible to create new data by time consuming manual segmentation which is usually supported by semi-automatic aids. However, the resulting data is still static geometry. In order to make use of movable and deformable peripheral nerve cords data structures and algorithms must be designed and implemented.

In addition, for a plausible anatomical simulation of an articulated virtual humanoid, the nerve cords have to interact with other anatomical structures, i.e., move together in accordance and allow for functional simulations.³ To achieve this goal, nerves and other structures must be linked together. Defining these links in situ and performing docking tasks requires a modeling environment which, too, is presented in this paper.

Further author information: (Send correspondence to Sebastian Ullrich)
E-mail: s.ullrich@rz.rwth-aachen.de, Telephone: +49 241 80 29248

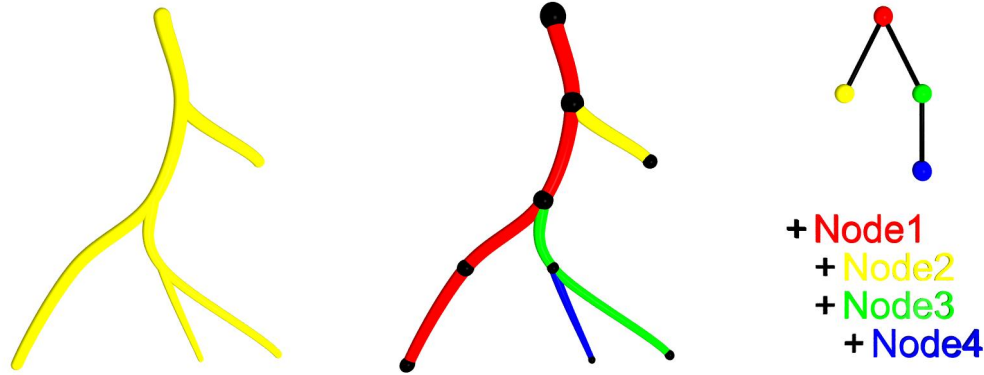


Figure 1. Example of our combined data structure for peripheral nerve cords: anatomical depiction of a nervous system on the left hand side, the according main hierarchical data structure for representations is shown in the middle, and functional tree data structure on the right side.

With this new system, nerves can be connected to bones or muscles for instance and will be therefore deformed analog to their connection. So individual models can be created, usable for example for surgical training in the RASim project (<http://www.rasim.info>), which focuses on a regional anesthesia simulator for medics. Other application areas are the anatomical education or the automatic segmentation, where missing nerves can be implemented afterwards to achieve better results.

2. METHODS

Our methods are divided into several parts. First, the main idea for the underlying data structure representing the nervous system is illustrated. The next subsection copes with building up this structure during a modeling process. At the end of this section, an idea of extending the flexibility by creating new datasets from existing ones and thus reducing the need for manual modeling is proposed.

2.1 Data Structure

To model and simulate peripheral nerve cords, we use a hierarchical tree data structure in combination with splines. Thus, each nerve section is represented by a spline curve and a node that can have several children. The spline curves are composed of control points which in turn are attached to movable virtual anatomical structures.

The data structure used for representing the virtual nerves is defined in two different levels (Figure 1). The abstract base level called `cord` deals with both geometric and functional characteristics shared by all anatomical cords/vessels. It consists of the three components `node`, `spline` and `intermediate point`. Each node features a list of splines whereat each spline in turn exhibits exactly two intermediate points. Due to the fact of each node having exactly one parent and one or more children, a hierarchical tree structure is built. The second level, the extension/derivation of the cord structure, is called `nerve`. Here, additional nerve-specific properties and operations retrofit the cord components. The `myoceptor`, a specialized intermediate point, serves as a link between a nerve and a muscle.

The data structure representing the whole nervous system is composed of a set of NerveSplines. These NerveSplines themselves contain NerveIntermediatePoints and NerveControlPoints. The NerveIntermediatePoints are the endpoints of the splines, which are always interpolated. In this way, the modeler is able to fix points where the nerve has to pass through. The NerveControlPoints are only approximated and provide further degrees of freedom for the modeling process.

Multiple instances of both types are either created by the user during the modeling process or predefined by specialized semi-automatic segmentation algorithms. The interpolation between two NerveIntermediatePoints is accomplished by using the NerveControlPoints as parameter for standard interpolation techniques on splines.

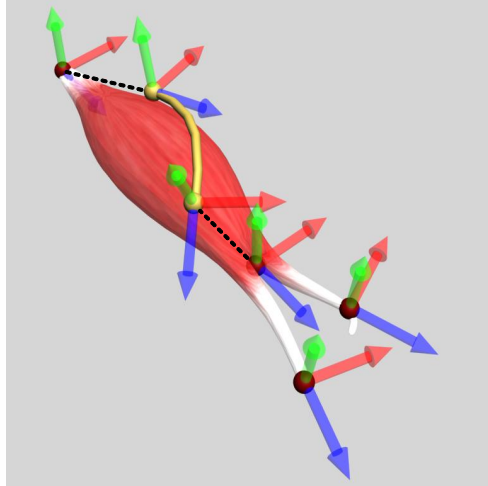


Figure 2. Example of the attachment of a nerve segment to an anatomical structure. The dotted lines illustrate which intermediate points and coordinate systems are linked together.

A NerveMyoceptor is used to provide an interface to the functionality of the corresponding muscle which it is directly attached to (Figure 2).

The NerveSplines are grouped together to hierarchical subsets by NerveNodes. Each NerveNode contains an arbitrary number of other NerveNodes representing the hierarchical layers beneath itself. The nodes of the resulting tree consist of sets of NerveSplines representing the smallest anatomical entities. Moreover, each NerveSpline is associated with an arbitrary number of muscles which are precalculated offline. These sets represent the muscles which are innervated once this nerve segment is triggered and starts to fire an impulse.

Due to the two-level design similar data structures for blood and lymph vessels can be easily created. In addition, the three anatomical classes feature a common base of properties and operations.

2.2 Simulation

The simulation and deformation of the nerve cords is achieved by interpolating splines. The control points are transformed accordingly to related anatomical structures. A movement of the limbs for instance induces deformation of the particular peripheral nerve cords. This deformation is done by readjusting the splines through interpolation during each update cycle. To save computation time the interpolation is constrained to segments that contain changed control points only. Although several interpolation schemes are possible,⁴ we chose Kochanek-Bartels splines due to their properties and intuitive parameters for modeling.⁵

2.2.1 Functional simulation

The hierarchical structure together with the relations between leaf nodes (nerve myoceptors) and muscles (innervation) allow for functional simulation, e.g. tracing a nerve impulse to a motor response. If a nerve segment is identified to fire an impulse, we have to determine which muscles are affected. In order to avoid traversal through the data structure at run time, we substitute this by some precalculation. This calculation provides a set of muscles for each single spline. These muscles are affected if the corresponding spline is triggered. First of all, each nerve myoceptor – located at a leaf of the node-tree – will be linked to the muscle which it is directly connected with. This is the initial state of the algorithm. The idea of the further proceeding is that a nerve spline, which is located in front of some other nerve splines (i.e., it is somewhere between this spline and the spinal cord) will affect all his successors by passing nerve impulses on to them and thus triggering exactly the muscles connected with all of them. To realize this main idea, we use the hierarchical structure provided by the nodes. The splines that are grouped together by the parent nodes will be associated with all muscles linked to their successors. We repeat this step until reaching the root of the node structure. This will lead to a mapping between splines and muscle sets. Splines within a specific node innervate the same set of muscles. Each of these sets matches the union of all sets of muscles associated with the splines that are united in successor nodes.

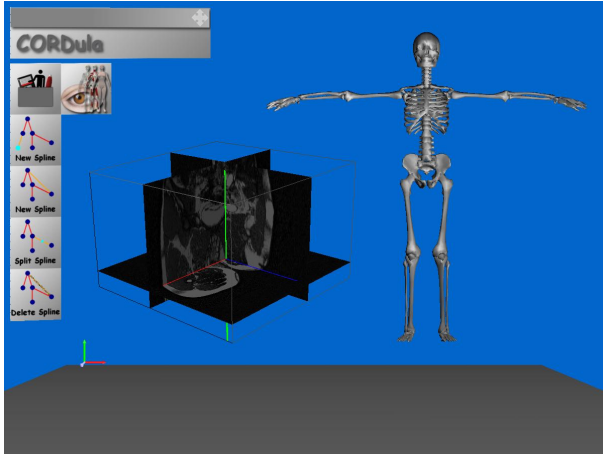


Figure 3. The modeling system allows for the interactive editing of control points. DICOM datasets can be co-located to verify the local anatomical setup.

2.2.2 Collision prevention

Nerve-nerve collisions and collisions with other anatomical structures can be usually ignored for the following reason. Since some intermediate points are attached to other anatomical structures, they move simultaneously so that self-intersections are mostly avoided implicitly. Another issue is collision or proximity between nerve and foreign matter (a needle for instance). The nerves are represented and rendered as splines consisting of control points and a certain kind of basis function (depending on the type of spline used). Due to the fact that splines are a convex combination of their control points, the curve itself lies in the interior of the convex hull spanned by the control points. So, collision detection can take place by checking the foreign matter against the convex hull. If needed, this procedure can be done more accurately by virtue of subdividing the spline. Though creating more splines and even more control points, this will lead to a more precise approximation of the curve.

2.3 Visualization

For visualization as well as collision detection, we have also implemented several geometrical representations. Polygonal lines provide the fastest visualization, and thusly are well suited for fine nerve cords and quick renderings. Thicker cords are represented either by tessellated cylindrical tubes or by geometries with arbitrary cross sections. The latter are defined via intermediate points and are interpolated accordingly.

2.3.1 Handles, Guides and Primitives

For the modeling environment, we need graphical representations of the objects that can be modified by direct interaction. Thus, we represent intermediate points and control points by spheres, use special handles for tangents, and have line segments show the connectivity. Additionally, we can utilize splines that are updated/redrawn, while moving an intermediate point or a control point.

2.3.2 Splines

In this visualization, we represent the nerve cords by rendering the splines from the data structure. The drawing routine supports both lines and cylinders as primitives. These primitives are fitted in between two adjacent interpolated points for all sample points of a spline. On the one hand, lines are very easy to render and they, too, give a pretty good overview in a very dense tree of nerve cords. On the other hand, cylinders are used to visualize the thickness or the diameter of a nerve segment. Due to the use of OpenGL display lists, they are still very fast.

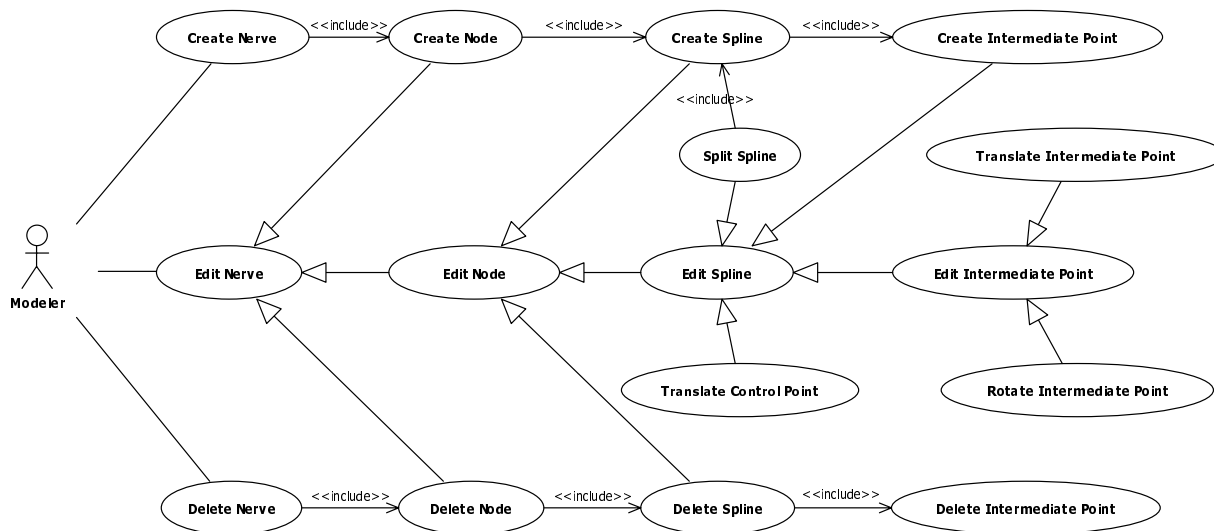


Figure 4. Overview of the possible actions within the modeling tool.

2.3.3 Tubes

We also provide a more advanced rendering technique to allow for better replication of the natural shape of certain nerve cords. For this purpose we use polygonal cross sections which permit representing rounded as well as flat nerve cords. The space between two adjacent cross sections is tessellated with a triangle mesh. During movement of the nerve cord the intermediate points of this mesh are transformed in accordance to the interpolated closest points on the spline curve.

2.4 Modeling

The modeling can be performed at a desktop system or in an immersive environment. The major feature of the system is the interactive placement of intermediate points for different anatomical structures with a real-time update of the involved data structures. First, an articulated virtual humanoid model with a skeleton geometry is loaded into the modeling environment. Then three-dimensional geometrical representations of anatomical systems and structures such as muscles, fat tissue and skin are applied to the model. To supply additional patient-specific information,⁶ DICOM datasets can be loaded and positioned interactively to be co-located with the geometry (Figure 3).

Some of the most important operations of the modeling system are creation and manipulation of *intermediate points*, specification of *branches*, *grouping* and setting of *names* (Figure 4). At the beginning of the modeling process, a NerveNode has to be created. All NerveSplines added to this node – usually those of a specific nerve – are grouped together and thus may be the target of a single operation such as deleting or disabling their subtree. A complex nervous system can be built by creating splines between an existing and a new intermediate point as well as between two existing ones. Creating and deleting a NerveIntermediatePoint on the other hand implicitly takes place when creating and deleting a NerveSpline, respectively. Intermediate point operations explicitly available to the user are translating, rotating and changing properties, e.g., the radius and the name. NerveControlPoints in turn can be translated only by what their appropriate splines will be affected.

In order to set the name of a nerve segment we make use of a functional anatomical ontology.⁷ The user thus can select the name at each branch from a limited amount of valid candidates.

2.5 Individualization

As a further tool to support the creation of new datasets, we propose a technique for synthesizing new datasets from existing ones (Figure 5). The intention is to create similar models that resemble different body types.

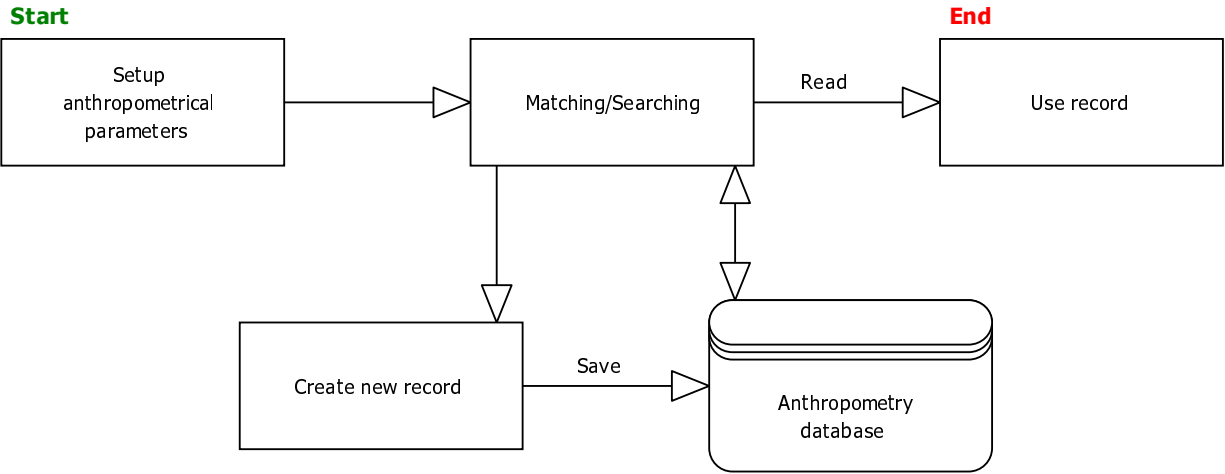


Figure 5. Illustration of the individualization process. Due to the specific parameters the database is searched for a match – and afterwards possibly updated – to readout the desired dataset.

To achieve this goal, we take anthropometrical parameters into account and thereby apply the appropriate transformations to bones and joints, to muscles, fat tissue and skin, and finally to the course of the nerves. The anatomical data of several individuals, including their nerve datasets, is stored in a database. For the synthesis we correlate this data with certain anthropometrical parameters.

As long as the body height of two different datasets is about the same, there is no need of readjusting bones and joints. Any muscle is connected to specific anatomical structures within its appropriate body. Since these connections – origin and insertion – are fix only the shape and the volume of the muscles have to be altered (Figure 6). Usually each nerve model is (multiple) connected to a muscle. Myoceptors are to transform according to their point of contact with the appropriate muscle. Intermediate points though have to be repositioned in respect of the volumetric transformatinos of the nearest muscle.

3. RESULTS

The developed algorithms for simulation and visualization of peripheral nerve cords are already utilized by a regional anesthesia simulator.⁸ The modeling tool is especially useful to add and adjust control points for

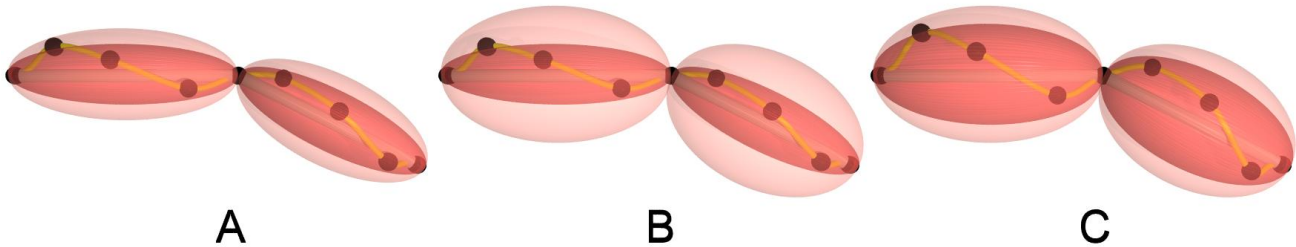


Figure 6. Example of individualization in respect of an arm. Starting with an ordinary figure (A) within the process we first get to a figure characteristic of overweight (B) and finally to the desired athletic figure. Apart from the skin all muscles (reddish ellipses) as well as nerve cords (yellow tubes) and its appropriate intermediate points (black spheres) are transformed.

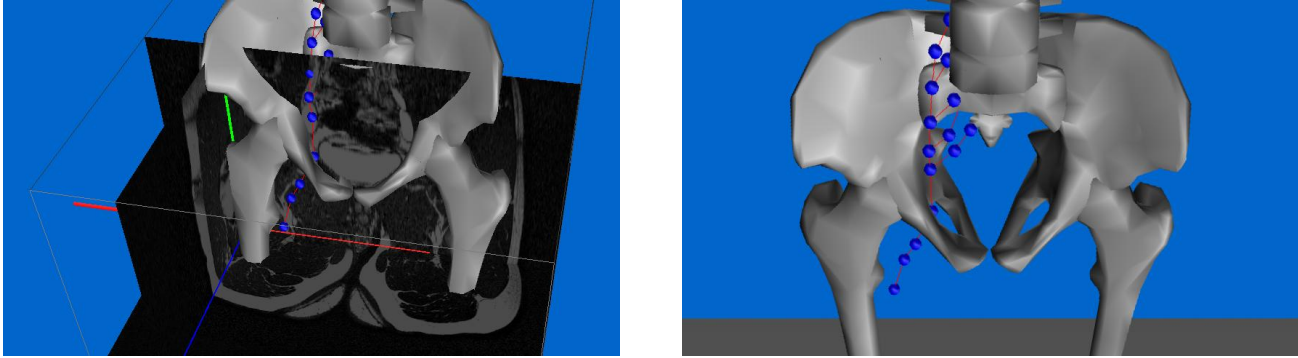


Figure 7. Example of the modeling of the sciatic nerve together with some reference material from a medical MRI scan.

segmented geometry to enable movable structures. The usability of a predecessor of the interactive modeling system that was focused on musculature has already been proven.⁹ Experts in the field of anatomy are currently using this novel system in combination with new acquired medical scans to create new three-dimensional datasets of peripheral nerve cords (Figure 7).

4. DISCUSSION

Until now, peripheral nerve cords have been mostly neglected in surgical simulation. With our approach creating new datasets by perturbing existing data sets and by convenient, intuitive modeling are both possible. Also rendering is based on well known techniques on splines and allows us to adapt the degree of detail depending on the requirements of the application. In addition, the data is not static but movable and deformable. Thus, it can be integrated into surgical simulation and also be used for teaching and understanding anatomy.

5. CONCLUSION

We proposed a hierarchical data structure to represent peripheral nerve cords in a virtual humanoid model. Simulation, deformation and visualization techniques have been implemented and allow for real-time interaction. An intuitive modeling system has been developed to create new datasets from scratch, add control points to existing static data (e.g., data from segmentation) or define inter-individual anatomical variations to create new subject-specific datasets. Future work includes the simulation of external electric stimulation of nerves as conducted during regional anesthesia procedures.

ACKNOWLEDGMENTS

This work was developed under the auspices of the German Research Foundation (DFG) grants no. KU 1132, LE 1108, RO 2000 and also partially supported by the START Research-Promotion Program of the Aachen University Clinic (UK Aachen). Furthermore, the authors like to thank Joel Mendoza and René Nissing for their individual contributions to the project.

REFERENCES

1. Interactive Complete Human Anatomy Series. ISBN 9781904369554. Primal Pictures; 2005.
2. Pflessner B, Petersik A, Pommert A, Riemer M, Schubert R, Tiede U, et al. Exploring the Visible Human's Inner Organs with the VOXEL-MAN 3D Navigator. In: Medicine meets Virtual Reality, Proc. MMVR 2001. IOS Press, Amsterdam; 2001. p. 379–385.
3. Ullrich S, Valvoda JT, Prescher A, Kuhlen T. Comprehensive Architecture for Simulation of the Human Body based on Functional Anatomy. In: Bildverarbeitung für die Medizin 2007. Germany; 2007. p. 328–332.
4. Lehmann TM, Gönner C, Spitzer K. Survey: Interpolation Methods in Medical Image Processing. IEEE Transactions on Medical Imaging. 1999;18(11):1049–1075.

5. Kochanek DHU, Bartels RH. Interpolating Splines with Local Tension, Continuity, and Bias Control. In: SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques; 1984. p. 33–41.
6. Teich C, Liao W, Ullrich S, Kuhlen T, Ntoubas A, Rossaint R, et al. MITK-based segmentation of co-registered MR data for subject-related regional anaesthesia simulation. In: Proceedings of SPIE Medical Imaging MI08; 2008. 6918 in press.
7. Rosse C, Mejino JLV. A reference ontology for biomedical informatics: the Foundational Model of Anatomy. *Journal of Biomedical Informatics*. 2003;36:478–500.
8. 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: *Bildverarbeitung für die Medizin 2007*; 2007. p. 202–206.
9. Valvoda JT, Ullrich S, Kuhlen T. Comparative Evaluation of User Performance for Modeling Tasks in Non-immersive and Immersive Virtual Environments. In: Proceedings of the IEEE VR 2005 Workshop New Directions in 3D User Interfaces. Bonn; 2005. p. 95–98.