



Nr.: FIN-004-2008

The Medical Exploration Toolkit

-

An efficient support for visual
computing in surgical planning
and training

Konrad Mühler, Christian Tietjen, Felix Ritter, Bernhard Preim

Arbeitsgruppe Visualisierung (ISG)



Fakultät für Informatik
Otto-von-Guericke-Universität Magdeburg

Technical Report

Impressum (§ 10 MDStV):

Herausgeber:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Der Dekan

Verantwortlich für diese Ausgabe:
Otto-von-Guericke-Universität Magdeburg
Fakultät für Informatik
Konrad Mühler
Postfach 4120
39016 Magdeburg
E-Mail: muehler@isg.cs.uni-magdeburg.de

<http://www.cs.uni-magdeburg.de/Preprints.html>

Auflage: 50

Redaktionsschluss: Juni 2008

Herstellung: Dezernat Allgemeine Angelegenheiten,
Sachgebiet Reproduktion

Bezug: Universitätsbibliothek/Hochschulschriften- und
Tauschstelle

The Medical Exploration Toolkit – An efficient support for visual computing in surgical planning and training

Konrad Mühler, Christian Tietjen, Felix Ritter, and Bernhard Preim

Abstract—Application development is often guided by the usage of software libraries and toolkits. For medical applications, the currently available toolkits focus on image analysis and volume rendering. Advanced interactive visualizations and user interface issues are not adequately supported. Hence, we present a toolkit for application development in the field of medical intervention planning, training and presentation – the MEDICALEXPLORATIONTOOLKIT (METK). The METK is based on rapid prototyping platform MeVisLab and offers a large variety of facilities for an easy and efficient application development process. We present dedicated techniques for advanced medical visualizations, exploration, standardized documentation and interface widgets for common tasks. These include, e.g., advanced animation facilities, viewpoint selection, several illustrative rendering techniques, and new techniques for object selection in 3d surface models. No extended programming skills are needed for application building, since a graphical programming approach can be used. The toolkit is freely available. Well documented interfaces facilitate the extension of the toolkit.

1 INTRODUCTION

Software assistants for intervention planning, e.g., for surgery, interventional radiology or radiation treatment planning are a relatively recent development. Surgical applications have special demands on visualization and interaction. It is not sufficient to display and analyze slice data and to create volume-rendered images. Instead, an in-depth analysis of the image data needs to be supported with appropriate 3D interaction techniques and advanced visualization techniques. With the MEDICALEXPLORATIONTOOLKIT (METK), we present a widely applicable library for application development that closes the gap between image analysis, processing, and basic visualizations on the one hand, and the surgical needs concerning visualization and interaction on the other hand. The METK is based on the image processing and development environment MeVisLab [MeVisLab, 2008] [Rexilius et al., 2006].

Visual computing in surgical applications requires to provide comprehensive patient related information, including visualizations of the relevant anatomic and pathologic structures, and enabling a faithful representation of the area around the pathologies. Moreover, measurements, annotations, resection lines and other information may be important to directly support pre-interventional decisions. On the one hand, flexible visualization and interaction is needed to cope with the peculiarities of individual cases, but on the other hand, strong guidance is desirable to avoid that surgeons are overwhelmed by the facilities. Most of these information, e.g., measurements of the structure's extent can only be derived after the segmentation of relevant structures. Visualizations based on pre-segmented structures are mandatory for operation planning in many fields, due to a high density of soft tissue structures with overlapping image intensity values. Thus, operation planning in the abdominal region (e.g., liver, pancreas or kidney), the neck region, and orthopedic interventions is preferably performed using isosurfaces in combination with the original 2d slices, while in neuro surgery or in emergency cases still volume rendering of the original image data is preferred. Therefore, the METK development focuses on surface-based visualizations of segmented structures. The METK does not support the segmentation process, since many established segmentation techniques are included in the underlying MeVisLab. Furthermore, several applications (e.g., HepaVision for liver surgery [Bourquain et al., 2002] or Neck-Vision for neck surgery [Cordes et al., 2006]) and service providers [MeVis Medical Solutions - Distant Services, 2008] are available to perform this task. The METK can import segmentation masks as well

as surface nets of structures (e.g., in Open Inventor or STL format).

Open source and freely available toolkits are widespread in the research application development. Using toolkits, application prototypes can be build up quickly, reverting on ready-to-use basic functions. In the medical domain, toolkits and libraries for image analysis and volume rendering are widely available [Caban et al., 2007], e.g., the MITK [Wolf et al., 2005], 3dSlicer [Pieper et al., 2004] or VolumeShop [Bruckner and Groeller, 2005]. However, they are difficult to program, since substantial C++ knowledge is required. The METK supports an easier application building process for surgical applications, where no extended programming skills are needed, since graphical programming in combination with script-based interface design and many visualization and interaction techniques (like automatic viewpoint selection and animations) are employed. The METK is a turn-key environment because all described functions are fully implemented and basic applications in form of example networks and datasets are provided with the METK.

The METK provides capabilities for application developers as well as techniques that directly support the end user of such applications – the surgeon. The METK differs from other toolkits in the following ways:

1. Providing a large variety of relatively recent visualization and interaction techniques as well as interface widgets and templates for surgical operation planning.
2. Faster building of ready-to-use applications with more user friendly interfaces than with most other available platforms, while no extended programming skills are needed.
3. The open source and well documented data interface enables developers to easily extend the METK by new modules.

Outline

In Section 2 we present different surgical application scenarios and derive requirements for surgical applications concerning visualization, interaction, case management, and interface design. In Section 3, we review related toolkits and work. In Section 4, we give an overview of the key concepts of the METK and present the techniques that we integrated in the METK. Furthermore, details with respect to new techniques for 2d overlay visualizations, of advanced facilities for selecting objects in a scene with multiple semi-transparent objects, and of facilities to reuse once defined visualization parameters are presented. In Section 5 we describe how application building is achieved using the METK and present some applications where the METK has been successfully applied. In Section 6 we close with a summary and a view on future developments.

-
- Konrad Mühler, Christian Tietjen, and Bernhard Preim, University of Magdeburg, E-mail: muehler@isg.cs.uni-magdeburg.de
 - Felix Ritter, MeVis Research, E-mail: felix.ritter@mevis.de

2 REQUIREMENT ANALYSIS

Our development is guided by experiences in several fields of surgical application development. Thus, we will present selected application scenarios and derive requirements for surgical applications and their development in this section.

Scenario 1: Neck dissections must be planned carefully. They are carried out for patients with malignant tumors in the neck or head region to remove lymph node metastases. Depending on the broadening of enlarged lymph nodes, only a few of them or a large part of the neck including muscles must be resected. A lot of structures has to be taken into account (e.g., vessels, muscles and up to 60 lymph nodes). The surgeons must explore the distances of all larger lymph nodes to vital structures of risk in order to judge if they are infiltrating vitally important structures or if there is enough space to safely resect them individually [Krueger et al., 2005].

Scenario 2: One application scenario in abdominal surgery is the resection of tumors in the liver, kidney or pancreas which are rather similar with respect to the demands on software support. Here, a tumor or several metastases need to be resected out of the liver volume with a specific safety margin. In difficult cases, this intervention requires an intensive and detailed computer-based operation planning process. The tumor and especially the surrounding vessels must be inspected carefully in 2d as well as in 3d. The remaining liver volume must be calculated with respect to vessel supply and drainage and under supplied tissue needs to be adequately visualized in 3d as well as in 2d [Preim et al., 2000].

Scenario 3: In spine surgery, small changes of the spine's anatomy can evoke symptomatic disorders for the patient. Hence, the spine surgeon must inspect the spatial relations between neural and spinal structures as well as the relation of the spine to surrounding muscles, vessels and glands. The surgeons need to place virtual needles and implants in the spine region to plan different strategies for the access route later in the intervention. Dedicated 3d visualizations can help the planning surgeons to find such access routes without injuring important structures.

For all scenarios, the exploration of the dataset must be as fast as possible in the clinical routine. Applications for presentations are of utmost importance for collaborative intervention planning like tumor board discussions, where a complex case is presented by one medical doctor to initiate an interdisciplinary discussion to finally come to treatment decisions. Due to the importance of such interventions for patients, their consultation is essential and special applications for demonstration purposes are crucial.

In general, surgeons are medical experts, usually with only modest computer experience. They benefit from faithful spatial renditions of the patient's individual anatomy, but they usually have no special abilities to explore and handle 2d and 3d data. From our experience with surgical applications, we derive basic requirements for such applications:

1. Surgical applications must primarily support the surgeon's decision-making process.
2. Measurement capabilities must be provided to support, e.g., distance, volume, and angle measurements, since these measurements are often closely related to surgical decisions.
3. Due to the importance of 2d slice data, 2d and 3d views of data should be coherent and synchronized, while the exploration of 3d data must be supported in particular.
4. Important structures need to be emphasized, preserving the context.
5. Dedicated techniques for special surgical fields should be provided (e.g., resection techniques for abdominal surgery, DTI visualization for neuro surgery, and multi-modal data visualization for cardio surgery).

6. Since surgeons use computers only occasionally, they dislike complex interfaces [Cordes et al., 2007]. Therefore, Commercial companies provide applications with carefully designed, highly visual user interfaces. Thus, pleasant user interfaces should be provided.

Besides those requirements that an end user application must fulfill, special requirements are necessary to support an efficient development process:

- Ready-to-use applications should be created quickly. Thereby, essential feedback of end users can be obtained in an early development stage.
- The application building process should not require extended programming, like implementing a lot of classes for application behavior and interfaces in C++.
- Existing applications as well as new applications should be able to be extended by new techniques, e.g., for special surgical requirements, obtained from feedback or techniques from newer visualization developments.

Even though there are many similarities of surgical applications, every field of use has its special needs. Nearly all surgical applications need an efficient case management and inter-application communication, advanced 3d visualizations of segmented structures and guidance for their exploration. Measurement facilities are less important for patient's consultation, while resection techniques are primarily necessary for abdominal surgery planning. Thus, a modular approach is desirable where only required features are integrated in an application.

3 RELATED WORK

Since medical visualization and medical image analysis is a very active field of research, several toolkits are available. We will discuss them in this section with respect to the presented requirements.

For the fast generation of visualizations, [Bavoil et al., 2005] presented VisTrails. VisTrails is a publicly available pipeline-based environment, where visualizations for many fields of use, like time-varying data or diffusion tensor, can be created. The system also provides the ability to reuse several parts of previous visualization pipelines using *visualization by analogy* [Scheidegger et al., 2007]. VisTrails focuses on the generation and exploration of singular datasets and does not support an application building process quite well.

VolumeShop [Bruckner and Groeller, 2005] is a standalone prototype for the interactive direct volume illustration of single datasets. Providing impressive facilities to create illustrations, it is very useful for presentation purposes, where an artist creates singular aesthetic visualizations. VolumeShop also enables the testing and development of new visualization techniques, because it is an open source system. But extensive programming skills are needed. The main drawback is the lack of support to build applications based on the presented visualization and interaction techniques.

The 3dSlicer [Pieper et al., 2004] was used in a variety of research applications, mainly in the field of neuro imaging. As the name implies, the focus of the 3dSlicer lies on slicing 2d volume data. It only provides some basic 3d visualization techniques.

The Medical Imaging Interaction Toolkit (MITK) [Wolf et al., 2005] is a C++ framework that is build up on ITK and VTK. It focuses on image analysis algorithms and interaction support for the segmentation and registration of medical images. The developers themselves state the MITK not intended to be '*an application framework*'.

JULIUS [Jansen et al., 2001] is a software framework that consists of a core application which can be extended by plug-ins. It focuses on the medical image analysis and provides algorithms for segmentation, registration and intra-operative navigation. For visualizations VTK is used, while no special visualizations for surgical applications were mentioned.

A C++ toolkit specializing in intra-operative support is the Image-Guided Surgery Toolkit (IGSTK) [Enquobahrie et al., 2007]. It supports the development of applications for interventional radiology procedures and image-guided surgery, where external tracking devices are applied.

A closed framework for the visualization of multi-modal medical images is presented by Manssour et al. [Manssour et al., 2001]. They focus on segmentation and registration, but do not demonstrate how individual applications can be created with their framework.

Another framework for medical image analysis is CAVASS [Grevera et al., 2007]. It provides simple volume rendering and iso-surface visualizations, while there is no dedicated support for application development, since CAVASS is a fixed system that is only extendable by its open source interface.

A recent development environment for fast visualization prototyping is the DEVIDE system [Botha and Post, 2008], which provides substantial capabilities in accessing and changing code and underlying dataflow networks during runtime. DEVIDE supports the development of new visualization and segmentation techniques well.

SciRun [Weinstein et al., 2005] is a problem solving environment that is based on dataflow networks consisting of single modules, e.g., for scalar and vector visualization, simulation, and image processing. Also end user applications can be created, called PowerApps.

Amira [Stalling et al., 2005] is a commercially available object-oriented extensible toolkit for scientific visualization. It provides a wide range of analysis, simulation and visualization techniques. Amira offers a visual programming approach where the dataflow metaphor is used to connect several modules to a network. Even though there are scripting and rudimentary GUI facilities available in Amira, one main drawback is the lack of support for application development. Using Amira, no end user applications can be created.

MeVisLab [Rexilius et al., 2006] is a freely available rapid prototyping platform with many similarities to Amira and a broad overlapping of functionality (see Bitter et al. [Bitter et al., 2007] for a comparison of MeVisLab, Amira, SciRun and MITK), since all offer visual programming and Amira and MeVisLab use Open Inventor. However, MeVisLab focuses on the medical domain and provides facilities for application development, like higher definition language to design application interfaces. ITK and VTK are also integrated in MeVisLab and a memory efficient way of handling large image data volumes is provided. Using a demand-driven scheduling, all calculations and time and memory expensive operations are only performed for the part of the data that is currently needed or visible. MeVisLab focuses on medical image analysis and provides an extensive library for those purposes. However, the provided features aim at handling single images and objects, but have no special capabilities to handle whole cases.

The METK reuses virtually everything from MeVisLab, including the MeVis library, the graphical network development paradigm and the efficient memory management for handling large volume datasets.

In addition to modular toolkits for individual setups, there are dedicated commercial volume rendering systems, such as Barco's VOXAR3D [Barco, 2008], the VolumeGraphicsLibrary [Volume Graphics, 2008] or Kitware's VolView [Kitware, 2008].

In essence, there are many supportive toolkits and frameworks for medical image analysis and visualization. But some of them are focused on the creation of singular impressive visualizations (e.g., VisTrails [Bavoil et al., 2005], VolumeShop [Bruckner and Groeller, 2005]), some of them focusing on medical image analysis (e.g., MITK [Wolf et al., 2005], 3dSlicer [Pieper et al., 2004]), and only some of them allowing to build up own standalone applications (e.g., SciRun [Weinstein et al., 2005], MeVisLab [Rexilius et al., 2006]). To the best of our knowledge there is no toolkit or framework to create efficient medical applications with high-end visualizations, adequate interaction techniques and user interfaces guidance.

A very short and preliminary report on the METK presented here will be presented as a software demo on a national workshop [Tietjen et al., 2008]. Techniques which have only mentioned in the short workshop paper are described here in detail for the first time.

4 THE ARCHITECTURE OF THE METK

In the MEDICALEXPLORATIONTOOLKIT each function is encapsulated in a module. Using MeVisLab's visual programming environment, modules can be freely combined in a dataflow network to build up individual applications with an individual feature profile. This allows the developer to design applications that support the individual workflow of different surgical intervention planning processes in an efficient and fast manner.

All functions are organized in three layers: the data management and communication layer, the visualization layer, and the exploration layer (see Figure 1). The lowest layer imports the case data and provides data management functions. The visualization layer comprises viewer classes and special rendering modules, whereas basic viewer classes and the volume rendering are reused from MeVisLab. All high-end interaction and exploration techniques that are necessary to create powerful surgical applications are available from the exploration layer. The layers and their provided functions will be described in the next sections.

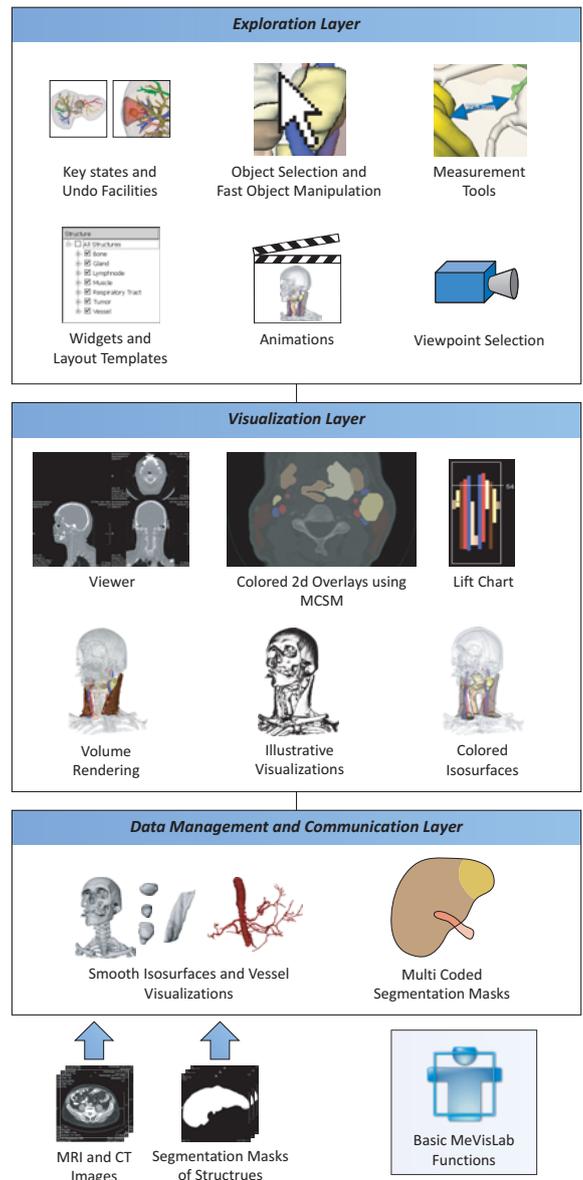


Fig. 1. **Layer structure of the METK.** A large variety of different visualization and interaction techniques as well as case management facilities and user interface widgets can be freely combined to build up individual surgical applications.

4.1 Data Management and Communication Layer

The database layer contains functions for case data management, inter-application communication and 3d isosurfaces generation. The METK can load and handle a wide range of medical CT and MRI image formats, including RAW and DICOM. Additionally, segmentation masks can be loaded to automatically generate 3d isosurfaces. This operation must be performed only once, since the isosurfaces are stored for further loadings of a case. Depending on the type of structures, different algorithms for generating the isosurfaces are used. In most cases, marching cube in combination with a constrained elastic surface smoothing, which was determined to be the best algorithm for most medical structures [Bade et al., 2007], is applied. For vessels we use a model-based surface reconstruction that respects the thin and branching structure of vessel trees [Hahn et al., 2001]. Besides segmentation masks, also surface nets of structures and secondary objects (like medical probes) can be imported. The METK provides facilities to convert cases segmented with other tools like ITK.

Case and cache management. To reduce the memory requirements and to speed up the whole process of loading and exploring a case, we integrated an efficient case management in the METK. Each structure and each image stack is only loaded once and distributed virtually in the application network. Even if the structures are visualized with different techniques in different viewers at the same time or if an image stack is browsed in a 2d viewer and a 3d viewer simultaneously (see Figure 8), it will only be maintained once in the application cache.

Communication and synchronization. Besides the data management, the METK provides a communication structure between all modules. Events can be sent between specific modules for a direct inter-module communication and be broadcasted to reach all modules. Therefore, all changes of underlying data and parameters are communicated to all modules 'listening' to those parameters, so they can adjust their own parameters, data, and visualizations. This leads to identical visual properties of all structures in all viewers and widgets and thus to a consistent view of all data.

Another information, communicated automatically in the METK, is the currently selected structure. Hence, a synchronized view in different viewers can be provided. If the user selects a structure in a 3d viewer, all 2d viewers can display the suitable slice for this structure and vice versa. If the user picks a structure in a 2d slice, it can be emphasized in all 3d viewers, moving the camera automatically to a good viewpoint on this structure (see Section 4.3).

The synchronization of parameters also enables the usage of synchronized 3d viewers. Different 3d viewers can be automatically synchronized in the METK by synchronizing its camera parameters (position, orientation etc.). This can be used to explore different datasets from the same viewpoints, to compare different intervention strategies for one patient or to compare pre- and post-operative data.

Multi-coded segmentations. Usually, each segmentation mask, representing one structure, is stored in a single file. This is inefficient in case of memory and performance. Storing all segmentation masks in one image stack can overcome this problem. However, one voxel of an image may belong to more than one structure, when structures overlap each other. For example a voxel in the liver tissue may belong to a tumor and the liver tissue itself. Thus, we cannot assign one label to each structure for the resulting segmentation mask of all structures. A straightforward approach is to assign each structure to one bit of a 8 byte voxel value. But this approach is limited by the number of bits: in our example, only 64 labels could be stored in a 8 byte value. Since in real data, only a small subset of all possible combinations of overlapping structures occur, we developed a more efficient solution: the multi-coded segmentation masks (MCSM).

An MCSM contains all segmentation masks of all structures of a case. Each combination of labels of a voxel that appears in the data is encoded with a distinct voxel value (see Figure 2). For example all voxels simultaneously belonging to the liver tissue and the hepatic vein (and to no other structure) are assigned one unique voxel value. The mapping of voxel values to structure lists is stored separately in the case data. An MCSM is created by sequentially adding one segmentation mask after another. If a new combination of voxel labels occurs

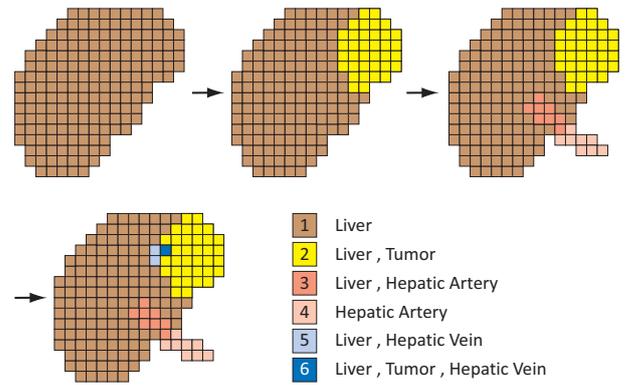


Fig. 2. **Multi-Coded segmentation masks.** Segmentation masks of single structures are sequentially added to the MCSM. For each structure all values of voxels that belongs to the structure are stored separately. Thus, many segmentation masks of overlapping structures can be saved in one MCSM.

at a voxel position, a new number is assigned to this combination. After all single segmentation masks were added to the MCSM, it can be used e.g., as an efficient base for colored overlays. The upper bound of 2^{64} labels will never be reached with medical datasets, since even the theoretical case that in a dataset of 512^3 voxels each voxel represents another combination of structures is covered by the MCSM.

4.2 Visualization Layer

In the visualization layer, all actions are performed that are necessary to provide basic and advanced 3d visualization techniques to the user.

Based on the isosurfaces stored in the cache, a material is assigned to each structure to achieve appealing isosurface visualizations. Important structures are visualized with a high opacity or with a silhouette. Thus, they are still visible as context structures, but do not hide the view onto important structures.

For **volume rendering** the METK employs the GigaVoxelRenderer [Link et al., 2006], which is integrated in MeVisLab. It enables the tagged volume rendering of segmented structures. Thus, different structures can be visualized with different transfer functions. By reason of consistency, the colors of structures are the same as in its isosurface visualization. To visualize unsegmented tissue, a global transfer function can be applied and the volume rendering can be combined with the isosurface visualizations.

Advanced 2d visualizations. The basic problem of the slice-based visualization, namely the lack of an overview in cross-sectional images, has been tackled with a 2.5D approach to provide the essential information, the LIFTCHART [Tietjen et al., 2006a]. A narrow frame attached next to the cross sectional image represents the overall extent of slices in the volume dataset. The top and bottom boundary of the frame correspond to the top and bottom slice of the dataset (see Figure 3). Each segmented structure is displayed as a bar at the equivalent vertical position inside this frame. Upper bars correspond to higher structures in the body. Different arrangements of the bars are possible, e.g., condensing all structures of the same type into one column. The currently displayed slice of the volume dataset is depicted by a horizontal line in the LIFTCHART widget (see Figure 3).

To support the correlation between structures in 3d scenes and 2d slices, structures can be visualized in 2d slice data as colored and semi-transparent **overlays**, so the underlying gray values are still visible. If more than one structure should be displayed at the same voxel position, the combined color can be calculated in different ways:

1. Only the color of the most important structure is chosen.
2. A weighted mixture of all colors of the overlapping structures is calculated.

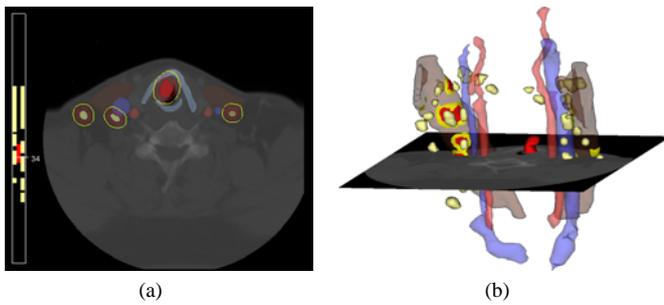


Fig. 3. **LIFTCHART**. 3(a) LIFTCHART in a 2d viewer and the corresponding dataset in 3d 3(b). The location of different structures in the slice stack can be identified by their color. Selecting a structure in the LIFTCHART selects the corresponding slice in the viewer. Furthermore, safety margins are depicted with red and yellow.

- Application-dependent overlapping regions can be emphasized separately in dependency of involved structures, e.g., the infiltration of lymph nodes in a muscle can be colored red with a silhouette, even if this visualization style does not appear in one of the two structures.

The calculation of the overlays is performed based on our multi-coded segmentation masks.

Safety margins are essential for the intervention planning and intraoperative navigation. Therefore for all structures at risk, a 3d Euclidean distance transform is performed. Depicting important distance thresholds (e.g., yellow and red) turned out to be appropriate. The distances may be displayed in 2d as well as in 3d. In the 2d view, silhouette lines as halos are drawn at the important distances. In 3d, unicolored surfaces are drawn on structures visualizing the range of distance to critical structures, e.g., lymph nodes to vessels and muscles (see Figure 3(b)).

Illustrative visualizations. As known from many anatomic books, context structures are mostly drawn stippled or with a silhouette. Those techniques were adapted for medical structures [Baer et al., 2007] and we provide it in the METK.

The visualization layer also provides several **viewers**. 2d viewers can display slices in many ways: singular or in a multi-slice view, where axial, sagittal, and coronal as well as free multi planar reformations can be shown. 3d and 2d viewers can be freely combined in an arbitrary number and arrangement.

4.3 Exploration Layer

To support the exploration process, we provide several techniques, interaction facilities and interface widgets.

Animation. To guide the user as well as to provide smooth transitions between different viewpoints and visualization styles, we integrated the animation framework developed by [Muehler et al., 2006]. Using an adaptive script language, one script with an animation description can be reused for many similar cases in an application, providing that segmentations results are stored and named in a standardized way. The script-based animations can also be used in an interactive application to guide the user's exploration process. Selected structures can be approached due to automatic camera flights, and appearance changes can be smoothly translated to preserve the user's orientation. The flexibility of the animation facilities enables the usage of advanced animation techniques like story telling principles [Wohlfart and Hauser, 2007]. Thus, applications based on the METK can provide both interactive animations in real time for exploration support, and rendered videos for presentation and interdisciplinary discussions like in a tumor board.

Viewpoint selection. Animations are enhanced by a dedicated viewpoint selection technique for multi-object 3d scenes [Muehler et al., 2007] that has been applied in several intervention planning tasks. Finding good views on single structures or groups

of structures is essential for an automatic guided exploration. After selecting a structure, the camera position is transformed automatically to a good view on this structure. The quality of a viewpoint is affected by many parameters. The structure should be visible to a maximum extend. A good viewpoint should be stable, i.e., minor rotations must not completely hide the selected structure. Medical doctors of different medical professions have different preferred regions to look at a 3d scene of segmented structures. Thus, the preferred region is also an important parameter for viewpoint estimation. These and other parameters are considered by our viewpoint selection technique. As discussed in [Muehler et al., 2007], relatively good presets for certain application scenarios may be defined. Thus, no further justification is necessary during the usage.

Good viewpoints are employed under several circumstances in the METK. They are used to generate standardized views for documentation (in combination with other standardized visualization parameters). If the user picks a structure from a list or from the viewer, the camera can be automatically moved to the best viewpoint of the structure.

Camera paths. To produce appealing and non-destructive camera paths by moving the camera from one viewpoint to another, we developed and integrated a set of different path algorithms in the METK. To preserve the orientation on long distance movements of the camera, we zoom out to a global view on the scene in the first part of the panning process and zoom into the target structure at the end of the flight. We also make camera movements more appealing by slow acceleration at the beginning and the end instead of abrupt speed changes.

Measurement tools [Preim et al., 2002] are also integrated, e.g., for distances measurements and its appropriate visualization by means of arrows. The proposed measurement tools are extended by automatic measurement facilities for computing minimal distances between two structures and by calculation of a structure's volume.

Key states. For presentation purposes, interdisciplinary discussions, or patient consultation several views and visualizations of the explored data needs to be saved. Instead of only saving screenshots, we introduce the concept of key states. In a key state, all information about a scene and its visualization is stored. This includes camera parameters as well as visualization properties. Thus, a complete state of a visualization can be restored later for further explorations or demonstrations. Since key states are stored in the case data, they can be transferred from one application (e.g., a surgical planning software) to another (e.g., an application for patient consultation). Naturally, key states can also be exported as screenshots for usage in documents or presentations.

Usually, a surgeon creates a couple of key states during a planning process (see Figure 4). In combination with our animation facilities, videos can be created automatically from a set of key states, where smooth transitions between the key states are computed automatically. Those videos are used to teach other surgeons for example. Key states can also be used to define presets. Applying a once defined key state to a new case with similar structures, those structures are visualized with the same properties as they where stored in the key state.

Undo. One important feature, especially for surgeons who are inexperienced in 3d exploration and mouse-based navigation, is an undo function for 3d scene manipulations. In the METK, after every performed action (e.g., a camera movement or a visualization change) the state of the whole scene is stored in a key state. Changes performed in a very narrow time range (e.g., automatic changes of the visualization) are combined in one key state. The user can return to arbitrary steps. In combination with the animation facilities of the METK, from a set of (undo) key states videos can be generated automatically. Those videos can be used for demonstration purposes or for the illustration of the exploration process for educational purposes.

Object selection. We provide several new techniques to select objects in 3d scenes with many objects of different transparencies. In such scenes the selection is ambiguous, if there is more than one object in the picking ray. The simplest approach to disambiguate the selection is always to select the first object in the pick ray. However, if this object has a high transparency, the user probably wants to select

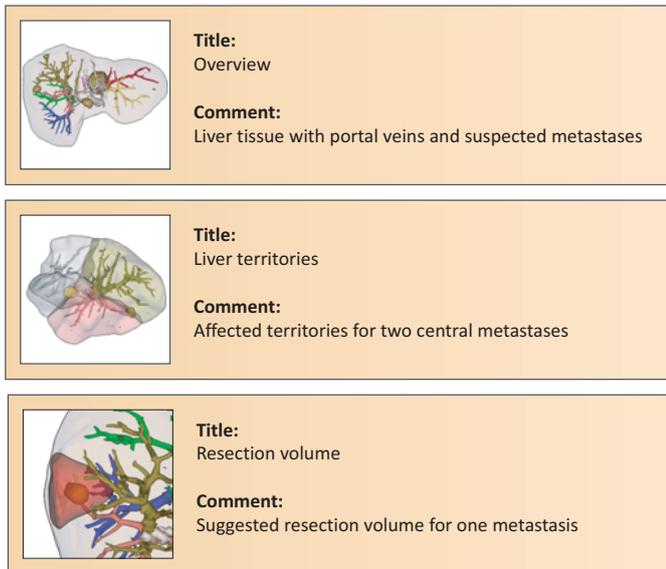


Fig. 4. **Key states.** Several key states that were created for planning a liver operation. The surgeon stored the key states together with a title and a short comment.

another object behind. In complex medical scenes, some objects are completely enclosed by others, so they are never the first object in the pick ray. For example in liver surgery, the liver tissue always encloses nearly all other structures like vessels and tumors (see Figure 5(a)).

We developed a procedure to automatically select an object, after the user clicked on the scene. It is assumed that the user points the mouse consciously. That means, when the mouse cursor is placed over a very large and a very small object, he or she placed it deliberately over the small object. Furthermore, it is assumed that the perception of very transparent objects is less than the perception of more opaque objects. To identify the desired object, the algorithm proceeds as follows: All objects hit by the pick ray are determined and sorted by the depth-distance of their intersection point. Only objects, which are locally visible by at least 10% are taken into account. As a next step, the size of the projected bounding box on the viewport of all objects and the transparency degree of the single objects is determined. The impact of transparency and the projected bounding box size are adjustable, to consider different types of scenes (e.g., scenes with objects of rather equal size). The object with the highest rating is selected at the end. Thus, for example opaque structures behind structures with a high transparency are selected. The algorithm reveals its limitations when a transparent structure is totally overlaid by an opaque structure with a similar size (see Figure 5(b)). In such cases, at all points the user picks the transparent structure, the opaque one in the background is selected. Therefore, we provide two interaction techniques in the METK. The first allows the user to scroll between all structures in the pick ray, using the mouse wheel or the cursor keys. Starting with the most presumably structure the user can scroll stepwise back and forth. The currently selected structure will be clearly emphasized, using a large silhouette and an opaque color. The second selection technique offers a list of all structures in the pick ray directly at the cursor position, like a popup menu, so the user does not need to defocus from the scene. We extend the list of textual structure names by pictorial representations of the structures. However, this technique aims at experienced users who know all structures by name (see Figure 6(a)).

Object manipulation. Even if users adjust the appearance of a visualization globally by selecting a preset, they might want to adapt the appearance of single structures individually. We provide some GUI widgets that can be integrated in a panel or window to adjust all visualization parameters like color, transparency or silhouette width. But we also provide an exploration technique, where the user can easily

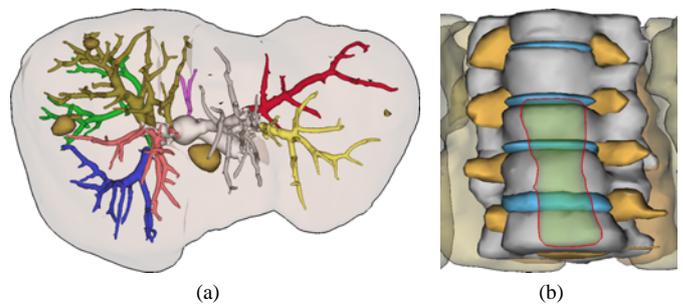


Fig. 5. **Object selection.** In 5(a) the selection of inner structures like vessels is made possible in the first place, while in 5(b) the transparent *oesophagus* in front of the opaque spine cannot be selected automatically.

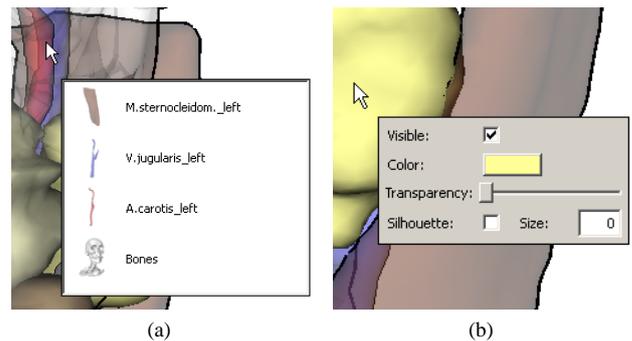


Fig. 6. **Fast exploration popup menus.** 6(a) All structures in the pick ray are presented as a fast accessible list to the user. 6(b) For quick parameter manipulation, a context popup is presented right to the cursor position.

adjust the most important parameters directly in the viewer. By right clicking on a structure, a popup menu appears at the cursor's position in the focus of the user and provides a slider for transparency, a color selector as well as the possibility to hide the picked structure. The provided list of parameters can be adapted and extended for individual application requirements. In some cases it might be useful to provide a button to request further information of the structure or a button to add this structure to a list (see Figure 6(b)).

Graphical application interface. For a fast and efficient application development, predefined widgets for common and recurrent tasks are provided, e.g., lists to select structures or to change their visibility. We provide panels to change the visualization parameters of structures like color, transparency or silhouette width, efficiently.

Feedback from surgeons using our applications clearly revealed that a rather low level of flexibility is needed and guidance is considered essential. Surgeons want clear and easy to understand interfaces [Cordes et al., 2007] instead of interfaces, overloaded with parameter sliders and value inputs. They want to get 'the best' visualization for the current task or medical question automatically or at the utmost selecting a well defined preset from a small list of choices. We support this, e.g., with our key states and the animation facilities. Furthermore, we also provide ready designed sample interfaces as templates for surgical applications that were approved by evaluation [Cordes et al., 2007] and by many interviews with our medical partners.

5 APPLICATION DEVELOPMENT WITH THE METK

Since the METK is an extension of MeVisLab, all METK applications can be build up by creating networks of modules in a visual programming environment. The interfaces are defined script-based. Thus, no extended programming skills are needed to create applications using

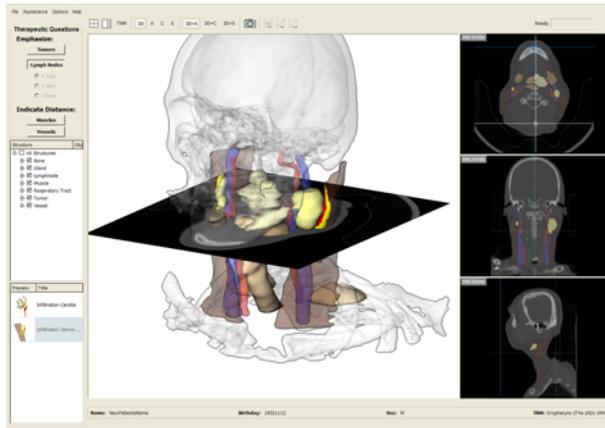


Fig. 8. **NeckSurgeryPlanner.** The NECKSURGERYPLANNER is an application to support the operation planning for neck dissections. To provide deep insight in the original 2d data as well as in the segmented 3d structures, 3d and 2d views are used synchronized. On the left, a browser for enabling and disabling structures and key state previews are provided.

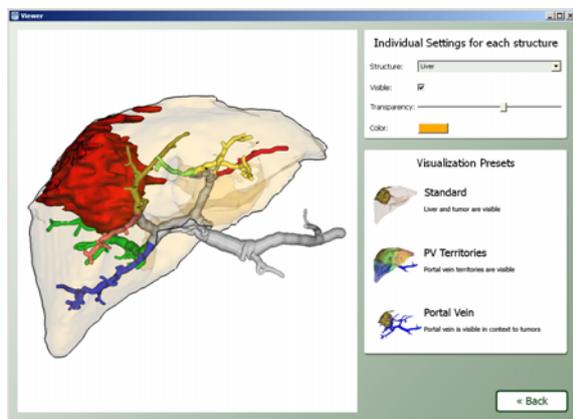


Fig. 9. **LiverSurgeryTrainer.** The LIVERSURGERYTRAINER is an application to teach abdominal surgeons the planning workflow for liver resection and living liver donor transplantations. The application layout contains only a few widgets.

the METK features. This quickly yields to applications that a surgeon can use (see Figure 7). Our experiences show that important feedback is given by surgeons only after they tried first functional prototypes. Thus, this stage should be achieved fast.

To extend the functionalities of the METK or to supplement existing functions, there are basically two options for developers: They can implement simple functions, like a patient data management or widget panels in modules, written in the script language Python. Advanced and especially performance critical issues can be implemented in C++ libraries. Since this is, for example, only necessary for new visualization techniques like DTI visualization, this does not contradict the supposed low programming skills that are needed to build up ready-to-use applications with the METK.

Several full-fledged applications were designed with our toolkit. Using the METK, a training system for liver surgeons was developed, the LIVERSURGERYTRAINER [Bade et al., 2006] (see Figure 9). The LIVERSURGERYTRAINER was evaluated in a comprehensive evaluation [Cordes et al., 2007]. The feedback from the surgeons inspired several refinements of the METK. For neck surgery planning the NECKSURGERYPLANNER was developed [Tietjen et al., 2006b] (see Figure 8). It supports the decision-making process for neck dissections.

6 CONCLUSION AND FUTURE WORK

We presented an extensive toolkit for surgical application development – the MEDICALEXPLORATIONTOOLKIT. It was shown that, using the METK, applications that fulfill surgical requirements of exploration support and visualization techniques, can be build up quickly. Introducing the multi-coded segmentation masks, we provide an efficient way to store multiple overlapping segmentation masks in one mask, supporting colored overlays in 2d. With our animation facilities, the viewpoint selection as well as our new support for object selection, we provide a substantial guidance for the exploration of 3d scenes.

Currently, we are working on bridging the gap between pre-operative planning software and intra-operative usage of the planning results. We plan to adapt selected techniques like the automatic viewpoint selection for an intra-operative use, since there more guidance in 3d exploration is needed, due to the special surrounding.

Our experiences with automatic techniques like the object and viewpoint selection showed that more semantic information about the importance of structures and their relations into the visualizations needs to be integrated. Therewith, the presented context to structures of interest or the intended user interactions can be adjusted in a more appropriate manner.

Due to the open structure and the free availability of the METK, many existing visualization and interaction techniques can be integrated, e.g., more sophisticated vessel visualizations [Schumann et al., 2007]. The METK does currently not provide special techniques for neuro surgery like DTI or for cardio vascular diagnostics like multi-modal visualizations.

One aim for the future is to detach the functionalities for animation, viewpoint selection and illustrative visualization to offer them to a wider community and other fields beyond surgical applications.

REFERENCES

- [Bade et al., 2007] Bade, R., Konrad, O., and Preim, B. (2007). Reducing Artifacts in Surface Meshes Extracted from Binary Volumes. *Journal of WSCG*, 15:67–74.
- [Bade et al., 2006] Bade, R., Riedel, I., Schmidt, L., Oldhafer, K. J., and Preim, B. (2006). Combining Training and Computer-assisted Planning of Oncologic Liver Surgery. In *Bildverarbeitung fuer die Medizin 2006 (BVM 2006)*, pages 409–413.
- [Baer et al., 2007] Baer, A., Tietjen, C., Bade, R., and Preim, B. (2007). Hardware-Accelerated Stippling of Surfaces Derived from Medical Volume Data. In Museth, K., Moeller, T., and Ynnerman, A., editors, *IEEE/Eurographics Symposium on Visualization (EuroVis)*, pages 235–242.
- [Barco, 2008] Barco (2008). <http://www.barco.com>.
- [Bavoil et al., 2005] Bavoil, L., Callahan, S. P., Crossno, P. J., Freire, J., Scheidegger, C. E., Silva, C. T., and Vo, H. T. (2005). Vistraills: enabling interactive multiple-view visualizations. In *Proceedings of IEEE Visualization*, pages 135–142.
- [Bitter et al., 2007] Bitter, I., Uitert, R. V., Wolf, I., Ibanez, L., and Kuhnigk, J.-M. (2007). Comparison of Four Freely Available Frameworks for Image Processing and Visualization That Use ITK. *Transactions on Visualization and Computer Graphics*, 13(3):483–493.
- [Botha and Post, 2008] Botha, C. P. and Post, F. H. (2008). Hybrid Scheduling in the DeVIDE Dataflow Visualisation Environment. In Hauser, H., Strassburger, S., and Theisel, H., editors, *Proceedings of Simulation and Visualization*, pages 309–322. SCS Publishing House Erlangen.
- [Bourquain et al., 2002] Bourquain, H., Schenk, A., Link, F., Preim, B., Prause, G., and Peitgen, H.-O. (2002). HepaVision2: A software assistant for preoperative planning in living-related liver transplantation and oncologic liver surgery. In Lemke, H. U., editor, *Computer Assisted Radiology and Surgery (CARS 2002)*, pages 341–346.
- [Bruckner and Groeller, 2005] Bruckner, S. and Groeller, M. E. (2005). Volumeshop: An interactive system for direct volume illustration. In C. T. Silva, E. Groeller, H. R., editor, *Proceedings of IEEE Visualization 2005*, pages 671–678.
- [Caban et al., 2007] Caban, J. J., Joshi, A., and Nagy, P. (2007). Rapid Development of Medical Imaging Tools with Open-Source Libraries. *Journal of Digital Imaging*, 20:83–93.
- [Cordes et al., 2006] Cordes, J., Dornheim, J., Preim, B., Hertel, I., and Strauß, G. (2006). Preoperative Segmentation of Neck CT Datasets for the Planning of Neck Dissections. In *SPIE Medical Imaging*. Spie Press.

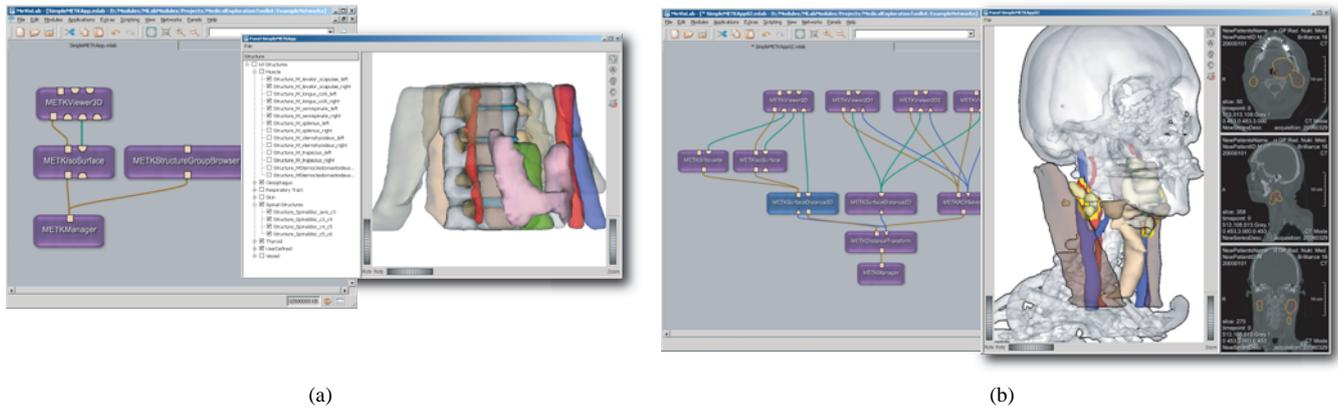


Fig. 7. **Sample METK networks applications.**7(a) A network to present segmented datasets in a 3d viewer with a GUI widget to change visibility of structures. A spine surgery dataset is displayed.7(b) A network of an application to synchronally explore 3d and 2d data, extended with silhouettes and a colored distance transformation in 2d and 3d. A neck surgery dataset is displayed.

- [Cordes et al., 2007] Cordes, J., Muehler, K., Oldhafer, K., Stavrou, G., Hillert, C., and Preim, B. (2007). Evaluation of a training system of the computer-based planning of liver surgery. In *Curac*, pages 151–154, Karlsruhe.
- [Enquobahrie et al., 2007] Enquobahrie, A., Cheng, P., Gary, K., Ibanez, L., Gobbi, D., Lindseth, F., Yaniv, Z., Aylward, S., Jomier, J., and Cleary, K. (2007). The Image-Guided Surgery Toolkit IGSTK: An Open Source C++ Software Toolkit. *Journal of Digital Imaging*, 20:21–33.
- [Grevera et al., 2007] Grevera, G., Udupa, J., Odhner, D., Zhuge, Y., Souza, A., Iwanaga, T., and Mishra, S. (2007). CAVASS: A Computer-Assisted Visualization and Analysis Software System. *Journal of Digital Imaging*, 20:101–118.
- [Hahn et al., 2001] Hahn, H., Preim, B., Selle, D., and Peitgen, H.-O. (2001). Visualization and Interaction Techniques for the Exploration of Vascular Structures.
- [Jansen et al., 2001] Jansen, T., von Rymon-Lipinski, B., Krol, Z., Ritter, L., and Keeve, E. (2001). An Extendable Application Framework for Medical Visualization and Surgical Planning. In *SPIE Medical Imaging*, pages 349–357, San Diego, CA.
- [Kitware, 2008] Kitware (2008). VolView. <http://www.kitware.com/products/volview.html>.
- [Krueger et al., 2005] Krueger, A., Tietjen, C., Hintze, J., Preim, B., Hertel, I., and Strauss, G. (2005). Interactive Visualization for Neck Dissection Planning. In *IEEE/Eurographics Symposium on Visualization (EuroVis)*, pages 295–302.
- [Link et al., 2006] Link, F., Koenig, M., and Peitgen, H.-O. (2006). Multi-resolution volume rendering with per object shading. In *Vision Modelling and Visualization*, pages 185–191. Aka GmbH Berlin.
- [Manssour et al., 2001] Manssour, I. H., Furuie, S. S., Nedel, L. P., and Freitas, C. M. D. S. (2001). A framework to visualize and interact with multimodal medical images. In *VG01 - Joint IEEE TCVG and Eurographics Workshop on Volume Graphics*, pages 385–398.
- [MeVis Medical Solutions - Distant Services, 2008] MeVis Medical Solutions - Distant Services (2008). http://mms.mevis.de/en/Distant_Services.html.
- [MeVisLab, 2008] MeVisLab (2008). <http://www.mevislab.de>.
- [Muehler et al., 2006] Muehler, K., Bade, R., and Preim, B. (2006). Adaptive script based animations for intervention planning. In *Proc. of Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 478–485. Springer.
- [Muehler et al., 2007] Muehler, K., Neugebauer, M., Tietjen, C., and Preim, B. (2007). Viewpoint Selection for Intervention Planning. In Museth, K., Moeller, T., and Ynnerman, A., editors, *IEEE/Eurographics Symposium on Visualization (EuroVis)*, pages 267–274.
- [Pieper et al., 2004] Pieper, S., Halle, M., and Kikinis, R. (2004). 3D Slicer. In *IEEE International Symposium on Biomedical Imaging: Nano to Macro*.
- [Preim et al., 2000] Preim, B., Selle, D., Spindler, W., Oldhafer, K. J., and Peitgen, H.-O. (2000). Interaction Techniques and Vessel Analysis for Pre-operative Planning in Liver Surgery. In *Medical Imaging and Computer-Assisted Intervention, MICCAI 2000*.
- [Preim et al., 2002] Preim, B., Tietjen, C., Spindler, W., and Peitgen, H.-O. (2002). Integration of Measurement Tools in Medical Visualizations. In *IEEE Visualization 2002*, pages 21–28, Boston.
- [Rexilius et al., 2006] Rexilius, J., Kuhnigk, J.-M., Hahn, H. K., and Peitgen, H.-O. (2006). An Application Framework for Rapid Prototyping of Clinically Applicable Software Assistants. In Hochberger, C. and Liskowsky, R., editors, *Informatik fuer Menschen - Band 1*, pages 522–528. GI-Edition - Lecture Notes in Informatics (LNI), P-93.
- [Scheidegger et al., 2007] Scheidegger, C., Vo, H., Koop, D., Freire, J., and Silva, C. (2007). Querying and creating visualizations by analogy. In *IEEE Transactions on Visualization and Computer Graphics*, volume 13, pages 1560–1567. IEEE Educational Activities Department.
- [Schumann et al., 2007] Schumann, C., Oeltze, S., Bade, R., and Preim, B. (2007). Model-free Surface Visualization of Vascular Trees. In Museth, K., Moeller, T., and Ynnerman, A., editors, *IEEE/Eurographics Symposium on Visualization (EuroVis)*, pages 283–290.
- [Stalling et al., 2005] Stalling, D., Westerhoff, M., and Hege, H.-C. (2005). Amira: A Highly Interactive System for Visual Data Analysis. In Hansen, C. D. and Johnson, C. R., editors, *The Visualization Handbook*, chapter 38, pages 749–767. Elsevier.
- [Tietjen et al., 2006a] Tietjen, C., Meyer, B., Schlechtweg, S., Preim, B., Hertel, I., and Strauß, G. (2006a). Enhancing Slice-based Visualizations of Medical Volume Data. In Santos, B. S., Ertl, T., and Joy, K., editors, *EUROVIS - Eurographics / IEEE VGTC Symposium on Visualization*, pages 123–130, Lisbon, Portugal. Eurographics Association.
- [Tietjen et al., 2008] Tietjen, C., Muehler, K., Ritter, F., Konrad, O., Hindenach, M., and Preim, B. (2008). METK - The Medical Exploration Toolkit. In *Bildverarbeitung für die Medizin (BVM)*, pages 407–411.
- [Tietjen et al., 2006b] Tietjen, C., Preim, B., Hertel, I., and Strauß, G. (2006b). A Software-Assistant for Pre-operative Planning and Visualization of Neck Dissections. In *CURAC*, pages 176–177.
- [Volume Graphics, 2008] Volume Graphics (2008). <http://www.volumegraphics.com>.
- [Weinstein et al., 2005] Weinstein, D., Parker, S., Simpson, J., Zimmerman, K., and Jones, G. (2005). Visualization in the scirun problem-solving environment. In Hansen, C. and Johnson, C., editors, *The Visualization Handbook*, pages 615–632. Elsevier.
- [Wohlfart and Hauser, 2007] Wohlfart, M. and Hauser, H. (2007). Story Telling for Presentation in Volume Visualization. In Museth, K., Moeller, T., and Ynnerman, A., editors, *IEEE/Eurographics Symposium on Visualization (EuroVis)*, pages 91–98.
- [Wolf et al., 2005] Wolf, I., Vetter, M., Wegner, I., Boettger, T., Nolden, M., Schoebinger, M., Hastenteufel, M., Kunert, T., and Meinzer, H.-P. (2005). The medical imaging interaction toolkit. *Medical Image Analysis*, 9:594–604.