A generic computer assisted intervention plug-in module for 3D Slicer with multiple device support

Release 1.00

András Lassó[1], Junichi Tokuda[2], Siddharth Vikal[1], Clare M Tempany[2], Nobuhiko Hata[2], Gabor Fichtinger[1]

July 10, 2009

[1]Queen's University
[2]Brigham & Women's Hospital

Abstract

Various frameworks and toolkits have been proposed for rapid development of computer assisted intervention (CAI) software. In this paper, we investigate how the open-source 3D Slicer application framework can be used for this purpose. We defined general requirements for CAI software to be able to evaluate and enhance 3D Slicer for interventional applications in general, and not just for a specific system. 3D Slicer is found to be an appropriate basis for CAI software, as its built-in functions fulfill many requirements and missing functionalities can be conveniently added. We describe the implementation of a CAI software based on extending core 3D Slicer functions. Three enhancements are described in detail: the management of workflow, DICOM image transfer, and multiple views. The resulting software fulfills general CAI requirements and supports two different MRI-guided prostate biopsy systems (each with a different imaging mode, robotic manipulator and calibration method), thereby demonstrating the usability of our software for implementing CAI applications in general.

Contents

1 Introduction

Development of computer assisted intervention (CAI) systems is a rapidly evolving field, with lots of new methods, devices, and applications appearing day by day. CAI software components have to be built on reusable toolkits, frameworks, and open interfaces to keep up with the pace of changes.

To date, a number of free open-source software frameworks and applications have been developed in the research community of image-guided therapy. A group from Georgetown University has been developing a high-level, component-based software framework for image-guided surgery applications, named The Image-Guided Surgery Toolkit (IGSTK) [1]. The framework is built on top of Visualization Toolkit (VTK) and Insight Segmentation and Registration Toolkit (ITK) and it provides a set of APIs for various functionalities required in image-guided therapy such as visualization, image input/output, and tracking device connectivity. This highly modular architecture allows users to rapidly prototype navigation software for their clinical application and validate it. NaviTrack proposed by Von Spiczak *et al.* [2] is based on the similar concept, but provides Extended Markup Language (XML) interface that allows the users to configure data flow pipeline without coding. The SIGN framework [3] is another toolkit for rapid development of image-guided navigation software, with support for various device interfaces and workflow management. The Surgical Assistant Workstation (SAW) provides a framework targeted for the daVinci surgical robot, but its generic design allows its usage with other telesurgical robot systems. Besides those software frameworks specialized for image-guided therapy, several medical image processing and visualization software applications are extended for image-guidance of surgical procedures. Medical Imaging Interaction Toolkit (MITK) is also a framework build on top of VTK and ITK and can be extended for image-guided therapy application with MITK-IGT component [4]. Another approach is to interconnect medical image processing and visualization software with existing surgical navigation systems. Papademetris and his group developed a network protocol to interconnect their research software, BioImage Suite, with a commercial neurosurgical navigation system (VectorVision Cranial, BrainLab Inc.) [5]. The underlying idea for this work is to investigate state-of-art image processing and visualization techniques, which are not available in the conventional navigation system in the operating room without any modification to the clinical system.

3D Slicer [6] is one of those free open-source medical image visualization and processing applications that have been investigated for surgical navigation. 3D Slicer was used as prototype software for neurosurgery [7], prostate [8] and liver biopsy and treatment [9]. It offers functionalities useful for surgical applications, including various image processing and visualization techniques as well as network interface (OpenIGTLink [10]) for imaging and therapeutic device connectivity. Those functionalities are provided as plug-in modules, which users can fully control from 3D Slicer's integrated graphical user interface. For data management, 3D Slicer provides its own scene graph architecture called Medical Reality Modeling Language (MRML), where all data e.g., images, models and transforms, are accessed from those modules. This centralized data handling mechanism allows the users to perform complex tasks without writing code, by combined use of the many elementary processing tools available in the integrated environment.

However, the flexibility in choice of functionalities and data often misleads the users and let them fail to follow a certain workflow. Therefore, our challenge here is to provide a ready-to-use integrated graphical environment that strikes the balance between usability and flexibility of clinical configuration. Analysis of generic CAI software requirements show that there are a few other features that are not yet supported

by the current 3D Slicer core, such as the ability to receive images directly from imaging devices through DICOM network transfer and the possibility to display information on multiple screens in both the control room and operating room at the same time.

In this paper, we describe our new framework on 3D Slicer that can be used for implementing a wide range of CAI systems. Our previous works concentrated primarily on the mechanical engineering aspects of the CAI system development and did not provide vendor-independent strategy for software architecture and integration. The engineering contribution of our current work is the software architecture and tools to implement CAI software in 3D Slicer in general, and specifically for MRI-guided prostate intervention. Firstly, we compare capabilities of 3D Slicer to generic CAI requirements. Then we present the architecture of our software and the new workflow, image transfer, and multiple display management mechanisms with that the most important CAI requirements can be fulfilled.

2 Typical CAI application workflows

In this section we describe two different MRI guided prostate intervention systems that we built previously and specify a common workflow, which is valid for both of these two systems and in general for other CAI systems.

MRI-compatible robot system for transperineal prostate intervention

This system uses an actuated MRI-compatible manipulator to insert needles into the prostate through the perineum. Instant feedback is provided during the needle insertion by displaying semi-real-time two-dimensional MR images that are taken in a plane aligned to the current needle position. The three main components of the system are the needle placement robot, a closed-bore whole-body 3T MRI scanner (GE Excite HD 3T, GE Healthcare, Chalfont St. Giles, UK), and 3D Slicer with a custom module, as the user interface for the entire system ([11], Figure 1). All components are connected to one another via Ethernet, and communicating with each other using OpenIGTLink and DICOM protocols.

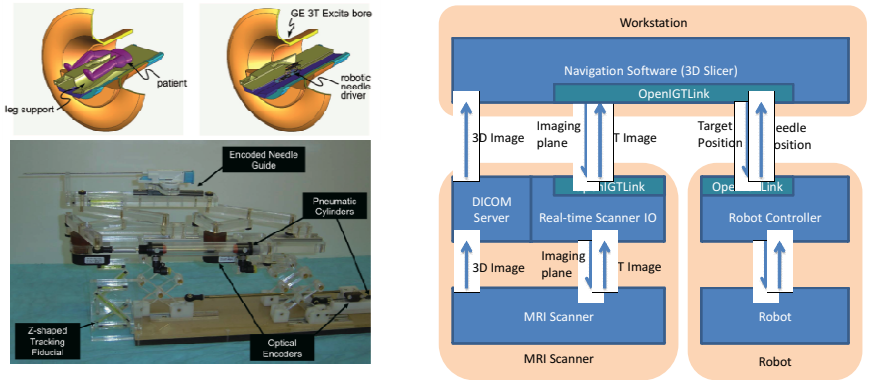

Figure 1. A robot for transperineal prostate biopsy and treatment (left) and its system configuration (right). Pneumatic actuators and optical encoders allow operating the robot inside a closed-bore 3T MRI scanner**.**

The system has six states, corresponding the phases of the clinical workflow: start-up, planning, calibration, targeting, manual, and emergency. In start-up phase system initialization is performed, including setup of software and hardware connections, sterile needle installation, and starting position adjustments. During the planning step 3D MR images are acquired, loaded into the prostate intervention module in 3D Slicer, and finally the target positions are defined on them. In calibration phase the robot coordinate system is registered to the image coordinate system by determining the position of the robot on the MR image from the Z-shaped fiducial that is attached to the robot base. In targeting state, for each planned target the robot is moved automatically to the desired position, while 2D images are acquired continuously to monitor the needle insertion. Manual and emergency states are used for positioning the robot by low-level motion commands, and stopping all robot motion for exceptional situations.

MRI-compatible robot system for transrectal prostate intervention

The system uses an MRI-compatible manipulator to insert needles into the prostate through the rectum ([12], Figure 2). MRI images are acquired before the needle insertion to define the targets, and after the insertion for verification. The end-effector is currently operated manually, by setting the targeting parameters that the navigation software has computed.

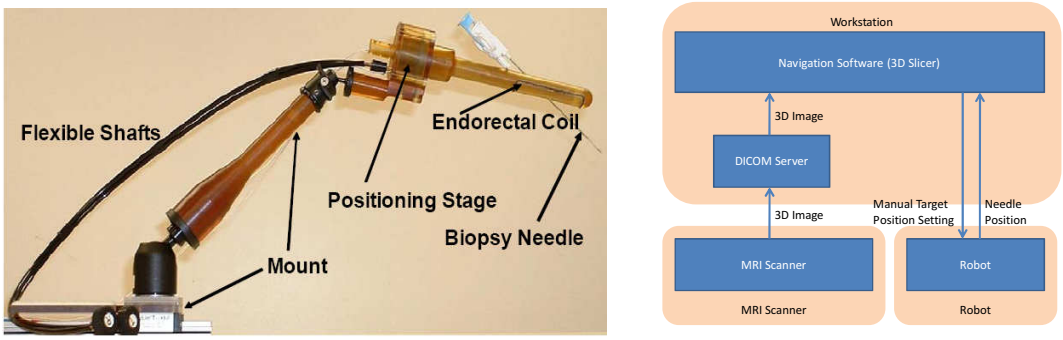

Figure 2. A robot for transrectal prostate biopsy and treatment (left) and its system configuration (right).

The navigation software has four states: calibration, segmentation, targeting, and verification. In calibration phase the robot coordinate system is registered to the image coordinate system, by locating on the MR image four markers that are attached to the manipulator. In the segmentation step a 3D model of the prostate can be created from an MR image for a better visualization of the relative positions of the targets, the needle, and the prostate. In the targeting step the user can define target positions and get the targeting parameters for a selected target. In the verification state an MR image is taken while the needle is inserted, and the distance of its actual visible position from the planned position is computed.

Generic CAI system workflow

By analysing the workflows of the above described two interventional systems, we can define the a common workflow, which is applicable to a both these two systems, and also to wide range of similar CAI systems.

- Start-up: In this state the necessary configuration parameters are set and software and hardware connections are established and tested.

- Planning: The intervention plan is created, describing what tools to be used and how, specifying target areas, positions, trajectories etc. If the planning procedure is lengthy or complex and information (images, measurements, other data) are already available pre-operatively then the planning is carried out before the intervention, while adjustments and simple planning entirely can be performed during the intervention.

- Calibration: In this phase the devices are calibrated and coordinate systems of interventional tools (such as the robotic manipulator), patient, and imaging device are registered to each other. This is completed as near as possible to the time of the intervention (i.e., right before the intervention or even during the intervention if that is practically achievable), to minimize the chance of any changes that could alter the calibration or registration.

- Targeting: The manipulator is moved to each planned position and the necessary operation (such as biopsy or seed placement) is performed. The users may monitor the progress of the operation by observing instant, continuous feedback of the system through displayed images and other measured parameters (tracked position, etc.).

- Verification: In this step the difference between the plan and the actual result of the intervention is compared. This information may be used for quality assurance and may trigger adjustments in the intervention plan.

- Manual Control: This state is necessary for all systems where the manipulator is motorized, to allow simple direct control the device in case of emergency or for testing purposes.

Some states that were present in the previously developed prostate intervention systems are removed because they can be considered to be part of these six states (segmentation is part of planning; emergency is part of manual control).

Although there is a natural order of states (Start-up, Planning, Calibration, then repeated Targeting and Verification), there may be a need to adjust the plan or calibration during the intervention, so generally any transition between the states shall be allowed at any time. It also means that any of the steps shall be allowed to be performed during the intervention.

3 Generic CAI software requirements

Functional requirements define what the software is supposed to do. They can be defined by analyzing the generic CAI workflow described in the previous section. In the Start-up state the software has to display controls on the user interface to configure the system and display status information. In the planning phase the software shall be able to load, display, analyze medical images and add planning information (such as target point positions). During Calibration and Targeting states the software shall be able to receive images, communicate with external devices (such as robot, MRI scanner) and visualize the images, tools. In Verification state receiving of the images and visualization is required. For Manual Control communication with external devices shall be supported.

Additionally, non-functional requirements, such as external interface requirements, performance requirements, design constraints, software system attributes shall be defined for the software ([13]). There are no strict constraints for external interfaces, however it is strongly recommended to use industry standard and/or open interfaces whenever it is possible, to avoid the overhead of implementing custom protocols. Performance (speed, response time, etc.) requirements and design constraints are not critical for prototyping software, so we do not define any generic requirement for them. However, there are important attributes that shall be considered if the software is to be used on patients: the safety, robustness and usability of the system shall be "good enough" to be efficiently used during interventions. Exact definition of what is good enough is a complex topic in itself (as discussed in e.g., by Di Maio et al. [14]), and shall be analyzed specifically for each application, but the following requirements are generally applicable. The software shall be able to recover from its failures by transitioning to a safe state where the procedure can be continued from, preferably by restoring a previous valid state. All significant information, such as user actions, state transitions, computation inputs and results shall be logged during the procedure to allow analysis of usage patterns, pro-active discovery of potential errors and investigation of failures after they have happened.

4 Implementation of a CAI application in 3D Slicer

The implementation of our MRI-guided prostate intervention software using 3D Slicer is presented in this section. Firstly, suitability of 3D Slicer as a basis for CAI software is discussed, then an overview of our software implementation is provided. Finally, three important extensions of the Slicer basic functionalities are described.

Suitability of 3D Slicer for implementing CAI software

Most of the functional requirements, such as the required image enhancement, analysis, segmentation, visualization features are nearly all implemented in the basic 3D Slicer application. Missing functionalities can be usually added easily, thanks to the modular architecture of Slicer and the rich feature set and flexibility of the underlying algorithm libraries (VTK, ITK).

3D Slicer stores all information that describes its state in an MRML scene, and this data can be saved at any time. By default Slicer saves this data only on user request, but to allow recovery from a software failure by reloading a previous state, this information can be saved automatically. An automatic save may be triggered by specific operations, such as the completion of important steps (e.g., as calibration or modification of the intervention plan), and before complex or resource-intensive tasks (e.g., loading or processing of image data).

Logging capabilities are already built into 3D Slicer, the CAI software just had to use this infrastructure by inserting code that reports all important actions to the logger during execution.

3D Slicer provides an integrated graphical user interface that allows the launching any of the numerous single-purpose software modules, in any order, on almost any data. All data, such as images, transforms and models are shared by all modules, this way the user can process, combine, analyze data by using different modules. However, as this process is not guided, and cannot be easily automated in Slicer, going through complex workflows takes time and careful, undivided attention of the user. This is hard to

guarantee during an interventional procedure, therefore the workflow and data flow management in Slicer needs improvement for CAI use.

Image guided navigation software require extensive capabilities for receiving images from other system components. 3D Slicer supports communication through the OpenIGTLink interface, which is targeted for efficient real-time communication with external devices and can be used to transfer any kind of information, such as images, position tracking information, or controlling commands. This is protocol works very well for those components that support it. However, the most widely supported method for medical image transfer through networks is DICOM [15]. The DICOM Working Group 24 has been even compiling surgical workflow models to determine the standard for integrating information about patient equipment and procedure [16]. Unfortunately, the current version of 3D Slicer does not have the capability to receive images directly through DICOM network transfer. An external application can be used to receive the images and then the images can be loaded manually into Slicer, but this manual procedure takes time and is error-prone. An extension shall be developed for CAI applications with that Slicer can directly connect to imaging devices and quickly and reliably receive the acquired images.

CAI applications generally require display both in the control room with lots of detailed information and controls, and a simplified display in the operating room with just the most important information in a very well visible format. As 3D slicer has no built-in support yet for managing multiple views on different displays, this has to be implemented as an extension for CAI applications.

Software implementation overview

3D Slicer is designed so that new functionalities can be easily added by implementing add-on modules. There are different types of Slicer modules, we chose to implement our CAI software in a “Loadable module”, because this integration method provides full access to Slicer internals (data structures, user interface, etc.).

The software runs on a separate workstation and communicates with the imaging device through Ethernet connection, using OpenIGTLink and DICOM protocols (Figure 3).

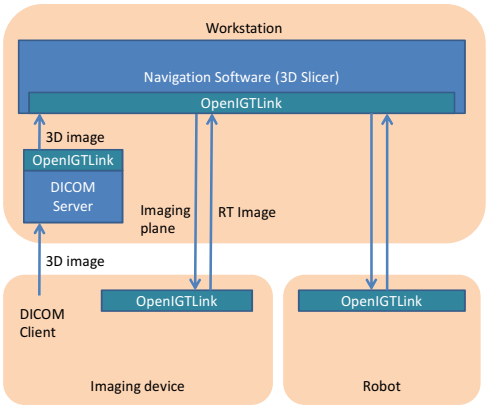

Figure 3. Architecture of the 3D Slicer based CAI software. Navigation software runs on a workstation that communicates with the imaging device and robot through standard communication protocols.

We created Slicer MRML node classes for storing all of our custom data: one node for storing all common data (configuration, OpenIGTLink connections, etc.) and a separate node for each robot type (one for transperineal, one for transrectal). We store images and target positions in standard Slicer nodes (vtkMRMLScalarVolumeNode and vtkMRMLFiducialListNode) and maintain references to them in our custom nodes.

Workflow management

Our CAI software provides a wizard interface for workflow management. The wizard interface consists of pages corresponding to clinical steps as we described in section 2, and shows only one page at once in the graphical user interface. This allows hiding unnecessary control widgets from the GUI, minimizing the risk of unwanted operation by the user. The steps are associated with states of software system, which defines the behavior of software components. The step transitions based on the operator's input on 3D Slicer and is notified to other software components through the network using OpenIGTLink protocol.

The user can configure the workflow, by ordering the existing components and defining MRML nodes shared by the components. To reuse the huge amount of useful software resources in 3D Slicer, it is important to incorporate the existing modules in the workflow interface. The mechanism to embed other modules into our workflow interface depends on the type of the modules. There are mainly two different types of plug-in mechanisms are used: a standard loadable (a.k.a. dynamic) module is loaded as a shared object or dynamic link library in 3D Slicer and has full control of 3D Slicer; Command Line Interface (CLI) is a mechanism to call a command-line program passing parameters as arguments and image data as files. Since a dynamic module creates its own graphical user interface (GUI) in certain area of the main 3D Slicer window, it is hard to embed the GUI in our wizard interface and restrict user input so that the users do not select wrong data. Thus, instead of displaying the module's original GUI, our CAI interface displays downsized GUI widgets that receive the user's input and internally call methods defined in the module. For CLI, the wizard displays forms for the users to input parameters and call the command with specified parameters. Data consistency is maintained by setting attributes of critical data so that they are cannot be manually edited in other modules (they are hidden).

Image transfer management

Our CAI software system offers two types of image transfer: DICOM image transfer from a PACS database or imaging device and real-time image transfer for monitoring procedures. Since OpenIGTLink interface is already available for transferring real-time image data, we provide proxy server software that converts DICOM image transfer to OpenIGTLink. The proxy server is implemented in two components. One component is listening at a TCP port to receive the images through DICOM transfer protocol and stores any acquired image in a file in a specific directory. The other component monitors the directory and sends any newly acquired image through OpenIGTLink connection. The advantage of this separation that the functionality of the first component is already available in some DICOM toolkits, so there is no need to implement it. We used the DCMTK [17] toolkit as DICOM receiver. Once the proxy software receives an image from through DICOM, it notifies 3D Slicer that the image is available for transfer with examination, series and image identifiers. The user can request the proxy to start transferring the image. For real-time image transfer, an OpenIGTLink interface has to be installed in the imaging scanner. We have developed a prototype of OpenIGTLink interface for GE's MRI scanner. The interface bypasses image database in the scanner system and push images directly to 3D Slicer through OpenIGTLink. 3D Slicer updates the display immediately after it receives the image. The interface can control the imaging

plane based on the transform matrix or quaternion received from 3D Slicer. Thus it is possible to acquire images from the plane parallel to the needle, by sending needle position and orientation measured by mechanical encoders on the robotic system.

Multiple views management

Displaying information in the control room and in the operating room is implemented by creating a viewer window that is independent from the Slicer main window and so can displayed on a secondary monitor (or projected into the operating room, Figure 4). The secondary viewer shall be flexible enough to display 2D slices, 3D objects and/or other simple text (messages, measurement results, etc.). The 3D viewer in Slicer (vtkSlicerViewerWidget) can display all these information, so we created a custom top-level window class and inserted an instance of the 3D viewer widget into that. The custom top-level window is implemented so that if a secondary monitor is available then it is displayed there (filling the full screen), otherwise it is displayed as a pop-up window that can be moved anywhere on the screen. If we simply used a second instance of the 3D viewer in the new window, the same information is displayed in both the Slicer main window and the secondary window, because the viewer widget instances share all inputs. This is good, because there is no need to redefine what and how should be displayed in the secondary view. However, certain information must not be shared between multiple viewers, such as the camera (because using the same camera in different windows can lead to infinite loop of updates and therefore software lock-up), and some other data (such as targeting parameters) may be required on one display only or display in a different style (e.g., text with larger, more legible font). To resolve these problems we created a 3D viewer widget that is subclassed from the original Slicer 3D viewer class, but some properties, such as the reference to the camera object and visibility of certain objects are overridden. Multiple camera and viewer support is already in Slicer's roadmap, and whenever it will be available, this secondary monitor implementation can be greatly simplified.

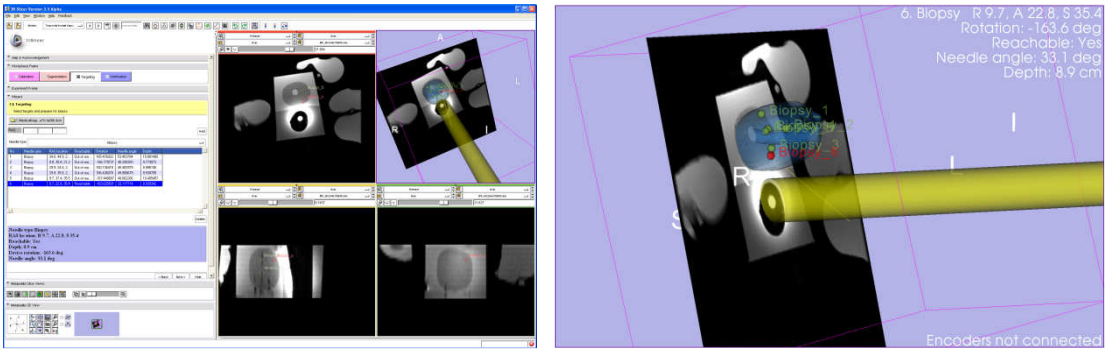

Figure 4. Example of a main 3D Slicer window (left) and secondary window (right). Compared to the primary window, the secondary window shows image and tools in larger size and additional targeting information (in the top right corner).

5 Conclusion

In this paper we presented a new computer aided intervention software application, based on the open-source 3D Slicer framework. We analyzed the most common requirements of CAI systems and then

utilized and extended the 3D Slicer core to comply with these requirements. The developed software can already be used to perform interventions with different MR-guided prostate biopsy systems, and we believe that the developed concepts and extensions can serve as a good basis for many other CAI software implementations.

Reference

[1] Andinet Enquobahrie, Patrick Cheng, Kevin Gary, Luis Ibanez, David Gobbi, Frank Lindseth, Ziv Yaniv, Stephen Aylward, Julien Jomier, and Kevin Cleary. The image-guided surgery toolkit IGSTK: an open source C++ software toolkit. J Digit Imaging, 20 Suppl 1:21–33, 2007. 1

[2] von Spiczak J, Samset E, Dimaio S, Reitmayr G, Schmalstieg D, Burghart C, Kikinis R. Device connectivity for image-guided medical applications. Stud Health Technol Inform. 2007;125:482-4.

[3] E Samset, A Hans, J von Spiczak, S DiMaio, R Ellis, N Hata, F Jolesz. "The SIGN: A dynamic and extensible software framework for Image-Guided Therapy", MICCAI workshop on Open Source and Data for Medical Image Computing and Computer-Assisted Intervention, Insight-Journal, DSpace Handle: http://hdl.handle.net/1926/207.

[4] Wolf I, Vetter M, Wegner I, Böttger T, Nolden M, Schöbinger M, Hastenteufel M, Kunert T, Meinzer HP. The Medical Imaging Interaction Toolkit. Med Image Anal. 2005 Dec;9(6):594-604

[5] Papademetris X, Vives KP, DiStasio M, Staib LH, Neff M, Flossman S, Frielinghaus N, Zaveri H, Novotny EJ, Blumenfeld H, Constable RT, Hetherington HP, Duckrow RB, Spencer SS, Spencer DD, S. DJ. Development of a research interface for image guided intervention: Initial application to epilepsy neurosurgery. International Symposium on Biomedical Imaging ISBI; 2006. p. 490–3.

[6] S. Pieper, M. Halle, and R. Kikinis. 3D slicer. 1:632–635, April 2004.

[7] Gering D.T., Nabavi A., Kikinis R., Hata N., O'Donnell L.J., Grimson W.E., Jolesz F.A., Black P.M., Wells W.M. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. J Magn Reson Imaging. 2001 Jun;13(6):967-75.

[8] Hata N., Jinzaki M., Kacher D., Cormak R., Gering D., Nabavi A., Silverman S.G., D A.V., Kikinis R., Jolesz F.A., Tempany C.M. MR imaging-guided prostate biopsy with surgical navigation software: device validation and feasibility. Radiology. 2001 Jul;220(1):263-8.

[9] Morikawa S, Inubushi T, Kurumi Y, Naka S, Sato K, Demura K, Tani T, Haque HA, Tokuda J, Hata N. Advanced computer assistance for magnetic resonance-guided microwave thermocoagulation of liver tumors. Acad Radiol 2003; 10(12):1442-9

[10] Tokuda J, Fischer GS, Papademetris X, Yaniv Z, Ibanez L, Cheng P, Liu H, Blevins J, Arata J, Golby A, Kapur T, Pieper S, Burdette EC, Fichtinger G, Tempany CM, Hata N, OpenIGTLink: An Open Network Protocol for Image-Guided Therapy Environment, Int J Med Robot Comput Assist Surg, 2009 (In Print)

[11] Fischer, G. S., Iordachita, I., Csoma, C., Tokuda, J., DiMaio, S. P., Tempany, C. M., Hata, N., Fichtinger, G. MRI-Compatible Pneumatic Robot for Transperineal Prostate Needle Placement, IEEE/ASME Transactions on Mechatronics, 2008, 13(3):295-30

[12] Krieger, A., Csoma, C., Iordachita, I., Guion, P., Singh, A. K., Fichtinger, G., Whitcomb. Design and Preliminary Accuracy Studies of an MRI-Guided Transrectal Prostate Intervention System Medical Image Computing and Computer-Assisted Intervention - MICCAI 2007, 10th International Conference, Brisbane, Australia, October 29 - November 2, 2007, Proceedings, Part II, Springer, 2007, 4792:59-67

[13] IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications, IEEE, 1998.

[14] DiMaio, S., Kapur, T., Cleary, K., Aylward, S., Kazanzides, P., Vosburgh, K., Ellis, R., Duncan, J., Farahani, K., Lemke, H., Peters, T., Lorensen, W., Gobbi, D., Haller, J., Clarke, L., Pizer, S., Taylor, R., Jr., Galloway, R., Fichtinger, G., Hata, N., Lawson, K., Tempany, C., Kikinis, R., Jolesz, F. Challenges in image-guided therapy system design. NeuroImage, 2007, 37:S144-S151

[15] The National Electrical Manufacturers Association (NEMA). Digital Imaging and Communication in Medicine (DICOM). NEMA Publications PS 31-318. 2007.

[16] Lemke H, Vannier M. The operating room and the need for an IT infrastructure and standards. International Journal of Computer Assisted Radiology and Surgery. 2006;1(3):117-21.

[17] DCMTK - DICOM Toolkit, version 3.5.4 (2005-12-20), http://dicom.offis.de/dcmtk