Integration of the Image-Guided Surgery Toolkit (IGSTK) into the Medical Imaging Interaction Toolkit (MITK)

Tong Lu · Ping Liang · Wen-Bo Wu · Jin Xue ·
Cheng-Long Lei · Yin-Yan Li · Yun-Na Sun ·
Fang-Yi Liu

Published online: 26 April 2012
© Society for Imaging Informatics in Medicine 2012

Abstract The development cycle of an image-guided surgery navigation system is too long to meet current clinical needs. This paper presents an integrated system developed by the integration of two open-source software (IGSTK and MITK) to shorten the development cycle of the image-guided surgery navigation system and save human resources simultaneously. An image-guided surgery navigation system was established by connecting the two aforementioned open-source software libraries. It used the Medical Imaging Interaction Toolkit (MITK) as a framework providing image processing tools for the image-guided surgery navigation system of medical imaging software with a high degree of interaction and used the Image-Guided Surgery Toolkit (IGSTK) as a library that provided the basic components of the system for location, tracking, and registration. The electromagnetic tracking device was used to measure the real-time position of surgical tools and fiducials attached to the patient's anatomy. IGSTK was integrated into MITK; at the same time, the compatibility and the stability of this system were emphasized. Experiments showed that an integrated system of the image-guided surgery navigation system could be developed in 2 months. The integration of IGSTK into MITK is feasible. Several techniques for 3D reconstruction, geometric analysis, mesh generation, and surface data analysis for medical image analysis of MITK can connect with the techniques for location, tracking, and registration of IGSTK. This integration of advanced modalities can decrease software development time and emphasize

the precision, safety, and robustness of the image-guided surgery navigation system.

Keywords MITK · IGSTK · Image-guided surgery navigation · Development cycle · Open-source software · Integration

Introduction

Minimally invasive procedures are becoming increasingly popular in today's healthcare system. The image-guided surgery navigation system enabled a surgeon to look through the anatomy of the patient, to see some instrument guided into the anatomy, and to plan the path of the instrument so that invasion is reduced and the accuracy of operation is improved. However, the development cycle of an image-guided surgery navigation system is too long to meet current clinical needs, and disease is too complex and diverse to use simple algorithms and functions for image processing and auxiliary guidance. Therefore, it is necessary to develop a software system that is able to shorten the development cycle and to ensure that the system is fully functional, conveniently interactive, and has high stability simultaneously.

With the development of open-source software, open-source toolkits for developing medical image processing applications played an important role in many research projects [1]. But at present, open-source software application in the medical field is far from meeting the actual requirements of the development for minimally invasive surgery, and very few people had tried to integrate two (or more) open-source software. In recent years, several benefits have contributed to the increased popularity and acceptance of open-source software, such as the high modularity

T. Lu · P. Liang (✉) · W.-B. Wu · J. Xue · C.-L. Lei · Y.-Y. Li ·
Y.-N. Sun · F.-Y. Liu
Department of Interventional Ultrasound,
Chinese PLA General Hospital,
28 Fuxing Road,
Beijing 100853, China
e-mail: liangping301@hotmail.com

of software functions, the use of new technology for the management and control of modules, more people testing to obtain more stable performance, etc. Thus, software system development using a variety of open-source libraries, especially integrating two (or more) open-source software, can shorten the development cycle while ensuring the comprehensiveness and stability of the system function.

The navigation of surgery means the use of geometrical and anatomical data to approach a target position in or at the patient. For that purpose, utilizing Image-Guided Surgery Toolkit (IGSTK) as a library is a good choice for the development of surgery navigation. This toolkit is based on open-source standards. IGSTK is a library that provides the basic components for navigation to enable researchers to develop a prototype rapidly and develop new applications for image-guided surgery applications [2, 3]. The main components include view, spatial objects, spatial object representations, image readers, registration, tracker components, etc. IGSTK was designed to interface tracking devices with software applications incorporating medical images, and it satisfied the requirement of key software design principles that emphasized abstraction, extensibility, reusability, fault tolerance, and portability for the image-guided surgery community. This library also follows a component-based architecture, with several components designed for specific sets of image-guided surgery functions. Because IGSTK was designed for safety-critical applications, the toolkit focuses on robustness. The software and documentation is freely available for download at <http://www.igstk.org>. However, the image processing function of IGSTK is too inferior to meet the needs of the image-guided surgery navigation system. Thus, the integration of IGSTK into another open-source software with powerful image processing functions is necessary.

Medical Imaging Interaction Toolkit (MITK) is one of the open-source software, and it has some powerful image processing functional modules with a high degree of interaction [4]. It is more important that MITK is a high-level, component-based, modular, cross-platform C++ application framework which provides a clear structure for the development of new applications and offers a plug-in mechanism to easily extend existing application [5]. The framework of MITK as well as the application can be easily extended by adding new transforms, metrics, and optimizers. Thus, developers of new algorithms are enabled to test and use their algorithms more quickly, spending less work on user interfaces [6]. MITK is available at <http://www.mitk.org>. MITK provided some very powerful functions for medical image processing and a framework with a clear structure; however, powerful, stable, and secure communication between a control computer and a tracking device was not provided. Image-Guided Therapy (IGT) is a navigation module of MITK, but it is too insecure and unstable to be

used in clinical surgery. IGSTK provides various tracking interfaces for various tracking devices, and its security and stability is superior to the IGT module of MITK. All of the features aforementioned are in line with the conditions of integration with IGSTK. We present an image-guided surgery navigation system by integrating IGSTK into MITK, and IGT was completely replaced by IGSTK. This system not only has a powerful medical image processing functions but also can provide the stable and secure communication between a control computer and a tracking device.

OpenIGTLink is an open, simple, and extensible network communication protocol for image-guided therapy to transfer transformed image and status messages. This protocol is feasible for the integration of devices and software. The protocol not only improves the interoperability of devices and software but also promotes transitions of research prototypes to clinical applications [7]. Therefore, OpenIGTLink has been adopted in this image-guided surgery navigation system for the integration of the two open-source software (IGSTK and MITK).

Methods

Use CMake to Connect MITK and IGSTK

The image-guided surgery navigation system uses MITK as a framework providing image processing, segmentation, and 3D visualization tools. IGSTK was used as a library that provides software interfaces for the electromagnetic trackers. These two open-source toolkits are both cross-platform, and both control themselves using CMake.

CMake is a cross-platform build system and supports multiple editors. It simplifies the configuration process by using platform-independent configuration files to generate native build files for the user-selected compiler. CMake automates the configuration process and makes it possible to develop a cross-platform toolkit without too much additional effort [8]. Therefore, it can connect MITK with IGSTK into an integrated functional system.

CMakeLists.txt file in the MITK library directory was modified to include the copied IGSTK libraries during the compilation of MITK. The IGSTK libraries consist of several subdirectories holding packages of algorithms. Since IGSTK uses CMake as a build system, each subfolder includes its own CMake configuration file. After compiling MITK, these CMakeLists.txt files get parsed recursively and include the complete folder tree of the IGSTK libraries to the build.

Establishment of Communication

Using the tracker component which was provided from IGSTK, the image-guided surgery navigation system can

handle communication between a control computer and track devices to obtain the position and orientation information for surgical instruments in the surgical scene. The position and orientation information had to be provided by identifying and recording fiducial points and the treatment path on the preoperative CT image using a mouse pointer. The corresponding fiducials in the physical body are identified using the tracker pointing device. A minimum of three non-collinear fiducial points are required.

This image-guided surgery navigation system had used three main classes to support the tracker:

1. IGSTK::Tracker

The class presents a generic interface for tracking devices to get position, orientation, and other relevant information from surgical instruments present in the scene. Derived subclasses provide tracker specific implementations for several widely used tracking systems.

2. IGSTK::TrackerTool

This class can track multiple tools simultaneously.

3. IGSTK::Communication classes

The communication class was implemented to establish the communication between the tracker class and the hardware tracking device.

Space Registration

At the core of the system is the space registration among preoperative CT image, the physical body, and the tracker pointing device to determine a relative relationship. The workflow of this application is outlined below:

1. According to the corresponding fiducials, the IGSTK::landmark3DRegistration class created a space transformation matrix.
2. The MITK::QmitkPointListWidget class was applied to identify and record four or more recorded spatial coordinates of fiducial points in the preoperative CT image.
3. Using the IGSTK::TrackerTool class, the system could identify corresponding spatial coordinates of fiducials in the physical body.
4. Based on the above two spatial coordinates, the space transformation matrix was calculated through the use of the IGSTK::landmark3DRegistration class.
5. The IGSTK::AuroraTrackerTool class will integrate the space coordinate of the tracker pointing device in the transformation matrix.

Integration of Two Open-Source Software (IGSTK and MITK) and Development of Functional Components

We integrated the proposed method to develop an image-guided surgery navigation system for the implantation of

clinical trials of radioactive particles and planning of microwave ablation therapies. The functional framework of this system is shown in Fig. 1.

Figure 1 indicates that this system can be applied to preoperative planning, intraoperative tracking, and postoperative evaluation. The keys of this system development were to integrate IGSTK into MITK; use magnetic locating device to obtain spatial coordinates of surgical instruments; and unify all coordinator information of instruments, images, and patient to achieve the same coordinate system.

As shown in Fig. 1, we use open-source software for a short development cycle and high stability of the image-guided surgery navigation system.

To effectively assist surgery, the set of functional components developed are listed below:

1. Medical image processing component

MITK provides some very powerful functions for medical image processing. Under the basis of the function modules provided in MITK, the image-guided surgery navigation system has developed a distinctive medical image processing component. By using the medical image processing functional component of this image-guided surgery navigation system, outline, segmentation, and 3D reconstruction of the structures of interest (related organs, vessels, lesions tumor) are performed. After image processing, surgeons can acquire the relative position and corresponding 3D geometric models of the surgery-related tumors, vascular, and related organs through the display of the computer.

2. System calibration and registration

Electromagnetic tracking is a promising navigation concept [9]. As a helpful tool in 3D navigation, aiming devices that are linked to the image-guided surgery navigation system allow for an accurate alignment of the trajectory and guidance of the PTC needle to the target [10–13]. The electromagnetic tracking system measures the real-time position of surgical tools and fiducials attached to the patient's anatomy. IGSTK was designed and developed to interface tracking devices with software applications incorporating medical images [14]. IGSTK provides interface functions for several commonly used tracking devices. By calling these functions of IGSTK, this system had implemented the calibration between the vertex of surgical instruments and the tracking tools and had realized registration among preoperative CT images, patient, and the surgical instruments. The purpose of calibration and registration is to achieve communication between the tracking devices and the control computer to get the position, orientation, and other relevant information from the surgical instruments present in real time. The calibration and registration of the coordinate system is shown in Fig. 2.

Fig. 1 Functional framework of this system

At first, by using a matrix method for coordinate transformation (MT), the coordinate information of the vertex of the “surgical instruments” and the coordinate information of the “tracking tool” had been united

(calibration). After uniting the “location system” coordinate information and the patient (phantom) coordinate information within the same coordinate system by using electromagnetic tracking devices (location system) and

Fig. 2 Calibration and registration of the coordinate system

the matrix method for coordinate transformation (MT), the united coordinate information was mapped to the preoperative CT image coordinate (registration). Therefore, the coordinate information of the vertex of surgical instruments could be acquired in the image coordinate at any time and could be displayed in three dimensions.

3. By uniting coordinate information, the integration of MITK and IGSTK has been generated.
 - (a) Coordinate information on the image in real time of the vertex of surgical instruments has been obtained in the `igstk::AuroraTrackerTool` class, and the space transformation matrix has been generated.
 - (b) By calling the `igstk::TransformObserver::GotTransform()` function, the space transformation matrix had been transformed in the `igstk::Transform` class.
 - (c) By calling the `Control6DTool` function of the `WxNavigationToolController` class, coordinate information on the image in real time of the vertex of surgical instruments was united with the coordinate information on the 3D image of the patient (phantom) which has been saved in the `mitk::BaseData` class.

The integration process takes a developer 7 days, and 52 code lines had been added in the system. The source code in the `WxNavigationToolController` class for the integration of MITK and IGSTK is shown in Fig. 3.

4. By calling the application programming interface (API) of IGSTK, this system had realized the real-time tracking of surgical instruments. An accuracy evaluation realized in simulated clinical conditions showed that the average tracking error of the needle tip was within 1 mm. Demonstration of the real-time navigation is shown in Fig. 4.

The 2D CT image of the patient (phantom) was shown in the figure (left). Using the yellow line, the edge of the tumor was outlined. The orange needle was shown as the planning path. The 3D images (patients (phantom), tumor, surgical instrument (real time)) were shown in the figure (upper right), and deviations between the planning path and the real-time 3D images of the surgical instrument were demonstrated in the figure (lower right).

5. The combination of modules

After the simulation of the validated calibration, tracking, and registration function of the image-guided surgery navigation system, the program developers had designed an interface component to guide surgeons. According to the recommendations proposed by surgeons with extensive clinical experience, developers had combined different modules to develop three functional modules (puncture, particle implantation, and microwave ablation). These functional modules will be appropriate for various clinical applications.

The Simulation Trials of the System

To demonstrate the system's efficiency, stability, and functional integrity, we presented several performance trials on an abdominal phantom in simulated clinical conditions.

In all simulation trials, we use a custom phantom electromagnetic tracking device (Aurora, Northern Digital, Waterloo, Ontario, Canada) and a computer (computer type: ThinkPad W510; CPU: Intel Core i7 Q720 1.60GHZ; RAM: 16 G; graphics card: NVIDIA Quadro FX 880M; OS: Windows7, 64-Bit) in the interventional radiology suite of the Department of Interventional Ultrasound, Chinese PLA General Hospital. Instruments are tracked by embedded 5 degrees of freedom sensors. These sensors report their

Fig. 3 Source code for the integration of MITK and IGSTK

Fig. 4 Image calibration, registration, and real-time navigation interface

pose up to an unknown rotation around the sensor axis. In our experiments, this is sufficient as the sensors are attached on the ultrasonic probe with the sensor and the tool axis coinciding.

At the same time, the simulation trials validated that using modular open-source software for secondary development can quickly modify and improve the system functionality to meet clinical needs. There were two surgeons (practitioners 1 and 2) with extensive clinical experience and two beginners (practitioners 3 and 4) taking these trials.

At first, in accordance with common medical needs, the developers combined the functional modules of the open-source software (IGSTK and MITK) to design three user-friendly interactive interfaces for surgery (puncture, particle implantation, and microwave ablation). Twelve user interfaces had been designed, and it took a developer 3 days; 568 code lines had been added into the system.

Secondly, based on the actual needs of clinical applications, according to the order from simple to complex, surgeons had designed three groups of simulation trials.

We use the following time line view to demonstrate these performances.

1. Image processing functions were provided by MITK, and the tumor had been manually segmented. The segmented tumor had been surface-rendered, and the CT

data of the phantom had been volume-rendered. All these 3D models can be shown/hidden for a 3D scene in the user interface.

2. According to the actual needs of the surgeries, practitioners had processed three-case puncture simulation trials, eight-case particle implantation simulation trials, and eight-case microwave ablation simulation trials. After completing the cases, practitioner filled the Test Tables carefully. In accordance with the table content, developers (developers 1 and 2) modified the software system and filled the relevant Modify Tables in a timely manner. Practitioners had assessed modified items by processing repetitive simulation trials seasonably.
3. Test tables were designed to improve this image-guided surgery navigation system for system stability, user-friendly interface, and functional integrity (Tables 1, 2, and 3). Modify tables were designed to record the number of modified items and modification time to evaluate the speed of modified system (Tables 4, 5, and 6).

Due to the operation of surgical puncture being a simple operation process, the system efficiency, stability, and functional integrity of the surgical puncture could be demonstrated by operating three-case puncture simulation trials. According to the planning path and the guided information shown in Fig. 4, practitioner 1 (surgeons) had finished the

Table 1 Test of simulation trials for surgical puncture

Practitioner	No. of system error messages	No. of computer crashes	No. of user interface items (proposed to modify)	Functions (proposed to modify)
Practitioner 1	0	0	4	Simplified procedures to improve the operation speed of navigation
Practitioner 2	0	0	0	None
Practitioner 3	0	0	0	None

Table 2 Test of simulation trials for particle implantation

Practitioner	No. of system error messages	No. of computer crashes	No. of user interface items (proposed to modify)	Functions (proposed to modify)
Practitioner 1	0	0	3	Add path planning (preoperative)
Practitioner 2	0	0	2	Add path recorded (intraoperative)
Practitioner 3	0	0	5	None
Practitioner 4	0	0	2	Add 3D information
Practitioner 1	0	0	1	None
Practitioner 2	0	0	0	None
Practitioner 3	0	0	0	None
Practitioner 4	0	0	0	None

operation process of single-needle surgical puncture. The surgeons proposed four user interface items that had not been used in the trial, and the speed of navigation would be affected for surgical puncture.

According to the proposal of practitioner 1, there were four user interface items that had been deleted and three related functions that had been modified. These modifications took developers half a day; 160 code lines had been deleted and 75 code lines had been modified in the system. After the integrated system had been modified, two trials were finished by practitioner 2 (surgeons) and practitioner 3 (beginner). These two practitioners both regarded this system as an efficient, stable, and fully functional navigational system for surgical puncture; there was no item that should be modified.

Due to the operation process of particle implantation being more complicated, eight trials had been operated for verifying the performance of the integrated system.

The first cases of the simulation trial of particle implantation had been performed by practitioner 1 (an experienced surgeon). The integrated system had been modified timely

by developers in accordance with the proposals put forward for particle implantation. Following the same process, the other three practitioners completed the simulation trials. In order to assess the performance of the system changes, the four practitioners operated the trials once again. There were 372 code lines deleted and 218 code lines modified for this group of simulation trials.

In the same process as the trials for particle implantation, eight trials for microwave ablation had been operated. There were 517 code lines deleted and 265 code lines modified for this group of simulation trials.

Results

By integrating IGSTK into MITK, we developed an image-guided surgery navigation system not only to provide very powerful functions for medical image processing but also to provide many functions for real-time medical instrument navigation. Based on the open-source software and following their Model–View–Controller (MVC) and state machine

Table 3 Test of simulation trials for microwave ablation

Practitioner	No. of system error messages	No. of computer crashes	No. of user interface items (proposed to modify)	Functions (proposed to modify)
Practitioner 1	0	0	12	1. Add safety margin planning (preoperative) 2. Add safety margin evaluation (postoperative)
Practitioner 2	0	0	6	None
Practitioner 3	0	0	3	None
Practitioner 4	0	0	0	None
Practitioner 1	0	0	2	None
Practitioner 2	0	0	0	None
Practitioner 3	0	0	1	None
Practitioner 4	0	0	0	None

Table 4 Modify simulation trials for surgical puncture

Developer	No. of functions (modified)	No. of user interface items (modified)	Modification time (days)
Developer 1	0	4	0.25
Developer 2	3	0	0.25

software design methodologies, the two developers only took 3 months to develop the software system prototype.

After the software system prototype had been completed, for the clinical practical value of the system, we had performed simulation trials for surgery (puncture, particle implantation, and microwave ablation) and had filled out tables and modified program timely to perfect the system.

The recorded contents of the test tables showed that:

1. The record number of system error messages and computer crashes had verified that this system has a high stability.
2. To meet the habits of different operators, the items of the graphical user interface (GUI) had to be modified in time.
3. To modify some functions and add special functions is the key of system development.

The recorded contents of modify tables showed that:

1. As a result of the adopted MVC and state machine software design methodologies of the open-source software (IGSTK and MITK) for secondary development, modification, and addition of the system, function modules had not impacted other modules and had ensured the safety and reliability, cleaner design, API simplicity, and consistent integration pattern, allowing quality control simultaneously. Because MVC and state machines software design methodologies can prevent the misuse of components and help manage complexity, traceability, and testing, bugs of the program had been reduced and the

Table 5 Modify simulation trials for particle implantation

Developer	No. of functions (modified)	No. of user interface items (modified)	Modification time (days)
Developer 1	0	3	1
Developer 2	3	0	3
Developer 1	0	2	0.5
Developer 2	2	0	1
Developer 1	0	5	1
Developer 1	0	2	1.5
Developer 2	5	0	2

Table 6 Modify simulation trials for microwave ablation

Developer	No. of functions (modified)	No. of user interface items (modified)	Modification time (Days)
Developer 1	0	12	2
Developer 2	7	0	15
Developer 1	0	6	1
Developer 1	0	3	0.5
Developer 1	0	2	0.5
Developer 1	0	1	0.5

change times and compilation times of the system had been shortened.

2. The items of the GUI provide the most straightforward visual feedback to clinicians to assist them in complete surgery efficiently. The trial results showed that there has been no impact on the function modules when these items had been modified.
3. Flexible modification and convenient management had confirmed that the integration of IGSTK into MITK can shorten the overall development cycle and adapt clinical needs in time.

Conclusion and Discussion

Trials of these simulation surgeries showed that in this image-guided surgery navigation system, accuracy of the surgical puncture had been improved, the phenomenon of re-puncture for position adjustment had been eliminated, and the overall time of surgery had been shortened. Especially for beginners, auxiliary effect is very significant. The most significant feature of this system is to adjust and modify quickly for clinical demand based on the modular mechanism in a timely manner.

Using this image-guided surgery navigation system, the related medical images had been processed for enhancement, segmentation, registration, and 3D reconstruction. According to the multimodal medical images, the system can analyze and integrate the 3D image information. The spatial information gained from 3D imaging is available in coordinates and thus forms an accurate input for performing the intervention. Stereotaxy enables a highly accurate probe positioning. Therefore, the characteristic of the image-guided surgery navigation system had been verified with regard to the security, stability, short development cycle, functional integrity, and so on.

An image-guided application that provides real-time visual feedback may assist clinicians in performing the procedure effectively and efficiently, as well as enabling them to target smaller lesions. With the increased popularity of minimally

invasive procedures in health care, the development cycle of the image-guided surgery navigation system has become increasingly in demand for medical applications. Oftentimes, application software development requires a lot of time and human resources. Using open-source libraries for the development of basic functionality can save resources by avoiding “reinventing the wheel” and can shorten the development cycle. A highly modular management is a significant feature of both IGSTK and MITK. Thus, it is possible to quickly adjust and modify based on the actual demand for software development. Using modular-based software, open-source software can also be tailored to meet actual demands in clinical trials of the image-guided surgery navigation system encountered and can be improved by the modular mechanism in a timely manner. The features mentioned are made to shorten the development cycle and improve the image-guided surgery navigation system performance as possible. Our integration of advanced modalities has decreased software development time and improved the precision, safety, and robustness of the image-guided surgery navigation system. The experiments indicate that the integration of IGSTK into MITK is feasible.

This feasibility study is very important for surgeons and software developers. The significance of scientific technology is that it can be fully utilized to be beneficial to mankind. Therefore, the integration of two (or more) open-source software will have revolutionary consequences for the development of open-source software.

Two frameworks (SOFA and VIMET) also provide excellent solutions for the simulation of medical procedures. Simulation open framework architecture (SOFA) is an open-source framework primarily targeted at medical simulation research. Based on advanced software architecture, it allows creating complex and evolving simulations by combining new algorithms with algorithms already included in SOFA [15]. The virtual medical training (VIMET) is a framework to build medical training applications by using virtual reality and a tool that helps the class instantiation of this framework. The main purpose is to make the building of virtual reality applications in the medical training area easier. The instantiation of the classes allows quick implementation of the tools for such a purpose, thus reducing errors and offering low cost due to the use of open-source tools [16]. It is necessary to study these open-source frameworks to enrich our integrated system for medical simulation research in future works.

The research directions of our future are different feature modules developed for the various needs of different clinical characteristics. A large number of animal experiments and

in vitro experiments will be performed to verify the functional integrity of the modules, the security and robustness of the system, convenient interaction with the interface, and the accuracy of navigation.

References

1. Wolf I: Toolkits and software for developing biomedical image processing and analysis applications. In: Thomas M Ed. *Biomedical Image Processing*. New York: Springer, 2011, pp 521–544
2. Cleary K; IGSTK Team: IGSTK: The Book, An Open Source C++ Software Library, 7 pp, 2007
3. Enquobahrie A, Cheng P, Gary K, Ibanez L, Gobbi D, et al: The Image-Guided Surgery Toolkit IGSTK: an open source C++ software toolkit. *J Digital Imaging* 20(Suppl 1):21–33, 2007
4. Wolf I, Vetter M, Wegner I, Böttger T, Nolden M, et al.: The Medical Imaging Interaction Toolkit (MITK): a toolkit facilitating the creation of interactive software by extending VTK and ITK. *Proc SPIE* 5367:16–27, 2004
5. Maleike D, Nolden M, Meinzer HP, Wolf I: Interactive segmentation framework of the Medical Imaging Interaction Toolkit. *Comput Methods Prog Biomed* 96(1):72–83, 2009
6. Stein D, Fritzsche KH, Nolden M, Meinzer HP, Wolf I: The extensible open-source rigid and affine image registration module of the Medical Imaging Interaction Toolkit (MITK). *Comput Methods Prog Biomed* 100(1):79–86, 2010
7. Tokuda J, Fischer GS, Papademetris X, Yaniv Z, Ibanez L, et al: OpenIGTLink: an open network protocol for image-guided therapy environment. *Int J Med Robot* 5(4):423–434, 2009
8. Martin K, Hoffman B: *Mastering CMake: a cross-platform build system*. Kitware Inc., Clifton Park, 2006, pp 1–3
9. Wood BJ, Zhang H, Durrani A, et al: Navigation with electromagnetic tracking for interventional radiology procedures: a feasibility study. *J Vasc Interv Radiol* 16:493–505, 2005 (April 4)
10. Bale R, Widmann G: Navigated CT-guided interventions. *Minim Invasive Ther Allied Technol* 16(4):196–204, 2007
11. Bale RJ, Hoser C, Rosenberger R, Rieger M, Benedetto KP, et al: Osteochondral lesions of the talus: computer-assisted retrograde drilling—feasibility and accuracy in initial experiences. *Radiology* 218(1):278–282, 2001
12. Bale RJ, Voge M, Martin A, Auer T, Hensler E, et al: VBH head holder to improve frameless stereotactic brachytherapy of cranial tumors. *Comput Aided Surg* 2(5):286–291, 1997
13. Widmann G, Eisner W, Kovacs P, Fiegele T, Ortler M, et al: Accuracy and clinical use of a novel aiming device for frameless stereotactic brain biopsy. *Minim Invasive Neurosurg* 51(6):361–369, 2008
14. Enquobahrie A, Gobbi D, Turek M, Cheng P, Yaniv Z, et al: Designing tracking software for image-guided surgery applications: IGSTK experience. *Int J Comput Assist Radiol Surg* 3(5):395–403, 2008. Nov 1
15. Allard J, Cotin S, Faure F, Bensoussan PJ, Poyer F, et al: SOFA—an open source framework for medical simulation. *Stud Health Technol Inform* 125:13–18, 2007
16. Gomes de Oliveira ACMT, dos Santos Nunes, FL: Building an open source framework for virtual medical training. *J Digit Imag* 23(6):706–720, 2010