

An open source software toolkit for image-guided surgery

Kevin Cleary, Luis Ibanez, David Gobbi, and Kevin Gary

A freely available software platform can help researchers rapidly and safely prototype systems for surgical applications.

Image-guided (also called computer-aided) surgery systems, were developed to provide physicians with a virtual, real-time display of the exact location of a surgical instrument relative to an anatomical structure (see Figure 1).¹ This interface enables the surgeon to work at precisely the site of interest while avoiding critical organs nearby. For the patient, image-guided surgery (IGS) entails substantially less trauma than open surgery, and systems are now commercially available for developing brain and spine procedures, among others. A software toolkit that facilitated rapid creation of such applications would be of considerable benefit to the active research community.

Each new image-guided application requires the creation of a great deal of software. A typical system consists of a three major components: a control computer, software for image processing and displaying the user interface, and a tracking system for locating the instruments and the patient in three-dimensional space. While hardware components can easily be procured as off-the-shelf items, the software that provides the functionality and usability for the application must be tailored to each new clinical application. Moreover, because the system will be used to guide surgical procedures, the software must be robust and reliable. Errors could lead to catastrophic results.

The image-guided surgical toolkit (IGSTK) was designed to meet these requirements in fulfilling the need for a software library capable of supporting safety-critical applications. The kit minimizes risk of error by incorporating a safety-by-design approach, achieved through the following principles. First, IGSTK uses a component-based layered architecture style. Every component has a well-defined set of features governed by a finite state machine (FSM). Strongly-typed interfaces provide enforceable interaction contracts between components. Second, the state of each component is explicit and always known, and all transitions are valid and meaningful. The state machine of each component is encapsulated: that is, clients of the component may not



Figure 1. In this image-guided system for brain surgery, the long slender bar at top right is an optical tracking system monitoring the position of the patient and the surgical instruments. A typical image display is at the far left. (Photograph courtesy of Richard Bucholz, MD, St. Louis University).

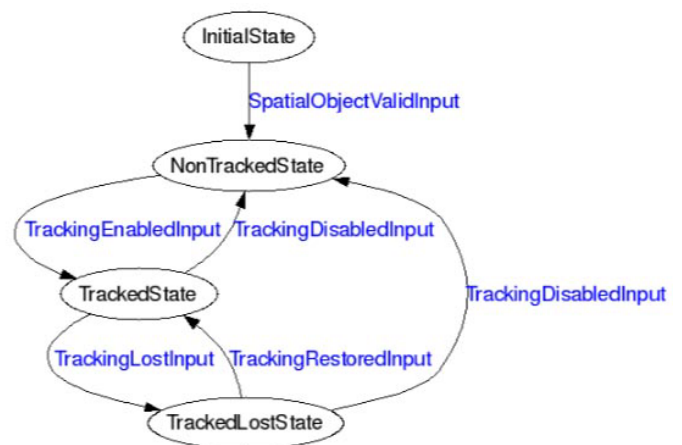


Figure 2. This scheme of a state machine for an IGSTK Spatial Object shows the four states in black and the transitions in blue.

Continued on next page

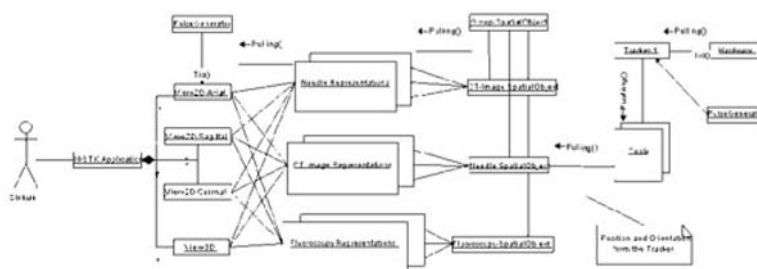


Figure 3. The IGSTK component architecture includes trackers, spatial objects, spatial object representations, and viewers.

manipulate the state outside the contract specified by the component interface.

Although state machines are not new to the literature, the use of their architecture for image-guided surgery is novel and a key feature of our toolkit.^{2,3} A state machine is defined by a set of states, a set of inputs, and directed transitions between states. State machines provide safety, reliability, and a consistent integration pattern. In addition, they facilitate quality control. We are currently developing automated methods, for example, to test state machine components. Figure 2 shows the state machine implementation for a spatial object component.

The IGSTK component architecture, as shown in Figure 3, includes several key software components. Trackers, first of all, provide the position and orientation of surgical instruments and tracking devices attached to the patient. Spatial objects provide the geometrical description of a surgical scene, while spatial object representations specify how objects should be displayed within the current surgical context. Finally, viewers form the user interface that presents renderings of surgical scenes to the physician.

The IGSTK project employs and advocates 'best practices' that include producing iterative releases, striving for 100% test code coverage, and the use of automated tools for testing and documentation.⁴ The development team consisted of 12 programmers who worked part-time over two years. The use of online collaboration tools such as a Wiki, mailing list, source code control, and bug tracking facilitated agility without sacrificing robustness.

In summary, IGSTK should enable biomedical research groups to rapidly create reliable prototype systems by providing software architecture and basic functionality. An initial release of the toolkit is available for free download, and details of the ongoing project can be found at <http://www.igstk.org>.

This research is supported by the National Institute of Biomedical Imaging and Bioengineering (NIBIB) at the National Institutes of Health (NIH) under grant R42EB000374 and by US Army grant W81XWH-04-1-0078. The content of this manuscript does not necessarily reflect the position or policy of the US Government. The authors would like to thank the IGSTK project team, including Rick

Avila, Stephen Aylward, M. Brian Blake, Patrick Cheng, Andinet Enquobahrie, Julien Jommier, Hee-su Kim, Sohan Ranjan, and James Zhang. The authors would also like to thank the IGSTK advisory board for advice throughout the project: Will Schroeder of Kitware; Ivo Wolf of the University of Heidelberg; Peter Kazanzides and Anton DeGuet of John Hopkins University; and Ingmar Bitter, Matt McAuliffe, and Terry Yoo of the NIH.

Author Information

Kevin Cleary

Imaging Science and Information Systems (ISIS) Center
Department of Radiology, Georgetown University
Washington, DC

Luis Ibanez

Kitware Inc.
Clifton Park, NY

David Gobbi

Atamai Inc.
London, Ontario, Canada

Kevin Gary

Division of Computing Studies
Arizona State University
Mesa, AZ

References

1. R. L. Galloway, *The Process and Development of Image-Guided Surgical Procedures*, *Ann. Rev. of Biomed. Eng.* **3**, pp. 83–108, 2001.
2. J. E. Hopcroft *et al.*, in *Introduction to Automata Theory, Languages and Computation*, 2nd ed., Addison-Wesley, 2000.
3. A. M. K. Cheng, *Real-Time Systems: Scheduling, Analysis, and Verification*, Wiley-Interscience, 2002.
4. K. Gary, M. B. Blake, L. Ibanez, D. Gobbi, S. Aylward, and K. Cleary, *IGSTK: An Open Source Software Platform for Image-Guided Surgery*, *IEEE Computer* **39** (4), pp. 46–53, 2006. doi:10.1109/MC.2006.130