Integration of a real-time video grabber component with the open source image-guided surgery toolkit IGSTK

Ole Vegard Solberg*a,b, Geir-Arne Tangen*a, Frank Lindseth*a, Torleif Sandnes*a, Andinet A. Enquobahrie*c, Luis Ibáñez*c, Patrick Chengd, David Gobbi*e,f, Kevin Clearyd

aSINTEF Health Research, Medical Technology and the National Center for 3D Ultrasound in Surgery, Trondheim, Norway;
bNorwegian University of Science and Technology (NTNU), Faculty of Medicine; Department of Circulation and Medical Imaging, Trondheim, Norway;
cKitware Inc., New York, NY, USA;
dImaging Science and Information Systems (ISIS) Center, Department of Radiology, Washington, DC, USA;
eAtamai Inc., London, Ontario, Canada;
fSchool of Computing, Queen's University, Kingston, Ontario, Canada

ABSTRACT

The image-guided surgery toolkit (IGSTK) is an open source C++ library that provides the basic components required for developing image-guided surgery applications. While the initial version of the toolkit has been released, some additional functionalities are required for certain applications. With increasing demand for real-time intraoperative image data in image-guided surgery systems, we are adding a video grabber component to IGSTK to access intraoperative imaging data such as video streams. Intraoperative data could be acquired from real-time imaging modalities such as ultrasound or endoscopic cameras. The acquired image could be displayed as a single slice in a 2D window or integrated in a 3D scene. For accurate display of the intraoperative image relative to the patient's preoperative image, proper interaction and synchronization with IGSTK's tracker and other components is necessary. Several issues must be considered during the design phase: 1) Functions of the video grabber component 2) Interaction of the video grabber component with existing and future IGSTK components; and 3) Layout of the state machine in the video grabber component. This paper describes the video grabber component design and presents example applications using the video grabber component.

Keywords: image-guided surgery, intraoperative imaging, open source software, ultrasound, video import

1. INTRODUCTION

For the benefit of the patient, systems for image-guided minimal invasive surgery and therapy are increasingly being used to safely navigate surgical instruments inside the human body. For visual feedback to the clinician, a graphical representation of the surgical tool is overlaid medical images (CT, MR, ultrasound, etc.) in much the same way as modern GPS systems overlaying a vehicle location onto a road map. It is therefore paramount that the medical images show an accurate picture of the current patient anatomy. Ideally, this would be based on an intraoperative real-time 3D image map considering influences such as respiration, pulsation, and surgical manipulation, which change the shape or location of anatomical structures during the procedure.

* Ole.V.Solberg (at) sintef.no; Telephone: +47 98245191; Fax: +47 93070800
The basic building blocks of a surgical navigation system are a computer, a tracking system, software, and optionally a medical imaging device that can generate real-time images. While tracking systems typically provide an application programming interface (API) for communication with the device, almost no real-time medical imaging devices offer something similar (i.e., an API for streaming the images from the device that generated them into the navigation computer). To provide the functionality needed for real-world clinical procedures, the navigation software can become fairly complex. At the same time, mission-critical applications like surgical navigation must be safe, robust, and accurate. Unfortunately, many resources within the IGS field are wasted in “reinventing the wheel” (e.g., code for interfacing with tracking hardware) instead of focusing on new research efforts. The open source image-guided surgery toolkit (IGSTK) attempts to address these issues by providing the basic functionalities needed for a navigation system. However, currently IGSTK does not have a component for streaming real-time image data into the system, and the paper presented describes an effort to add support for this feature.

1.1 IGSTK overview

IGSTK is an open source C++ library for building image-guided surgery applications [1][2][3][4]. The toolkit is developed with support from the National Institute of Biomedical Imaging and Bioengineering (NIBIB) at the National Institute of Health (NIH). Both industrial and academic partners have contributed to the development of the toolkit. The toolkit aims at providing basic components required to develop image-guided surgery applications. The initial version of the toolkit was released in February 2006 at the SPIE Medical Imaging conference at San Diego. Since then, applications built using the toolkit have been demonstrated at various scientific conferences (SPIE 2006, 2007 and SMIT 2007). Furthermore, an FDA approved single center clinical trial for electromagnetically tracked lung biopsy application developed using IGSTK has begun at Georgetown University Medical Center.

IGSTK follows a component-based architecture [1]. Each component has a well-defined set of behaviors governed by a state machine. Given that the implementation methods are correctly followed, the state machine will ensure that each component is always in a deterministic state and all state transitions are valid and meaningful. State machines were included as an integral part of the toolkit design with the purpose of producing a safe and reliable software library suitable for safety-critical applications.

Key components in IGSTK are views, spatial object representations, spatial objects, and a tracker component [1]. View objects render virtual representations of physical objects on the computer display. Spatial Object Representations are visual representations of Spatial Objects, which in turn are geometrical representations of physical objects in the surgical field. A pulse generator drives the update of the view, which sends requests to the tracker tool to get its latest spatial position (through the representation and spatial object). This ensures the accurate rendering of the surgical scene. Events are used for communications between different components. Response to a service request is usually in some form of event with or without data payload.

1.2 Real-time image acquisition overview

Previous versions of IGSTK only supported preoperative data. The new video grabber component will allow importing intraoperative video stream into IGSTK. To develop a video component, we chose to include support for analog video first, since there is no digital video standard that is supported across a broad range of medical devices. This video stream may consist of 2D ultrasound data or other video sources such as endoscopy video. Imported 2D ultrasound may be presented relative to other imaging modalities, such as preoperative CT or MR, or be processed further into 3D ultrasound [5]. The processing and presentation of the imported video is naturally dependent on the other IGSTK components. The video grabber component is one of several important steps in making IGSTK a complete toolkit for image-guided surgery.

The focus of the IGSTK video grabber component is the capture of an analog 2D real-time video stream, which is only one of many possible real-time imaging modalities. Figure 1 gives an overview of the IGSTK video grabber in relation to the other modalities. The digital branch will mostly consist of Software Development Kits (SDK) made for specific ultrasound scanners. These SDKs may provide either scan converted images or raw ultrasound data (radio frequency data).
2. VIDEO GRABBER COMPONENT

Currently the video grabber component is being developed by SINTEF, a Norwegian research foundation, in cooperation with the IGSTK development team. The IGSTK development process is based on an agile methodology [1]. For this component, we first brainstormed a list of requirements and made an initial design document. These were posted on a Wiki page and discussed both on the Wiki page and at bi-weekly teleconferences.

2.1 Requirements

The new video grabber component is based on the following list of requirements:

- Import real-time video without noticeable delay.
- Synchronization with other IGSTK components.
- Support cross platform development for portability
- Ensure operability with hardware used by IGSTK partners.
- Grab video stream and single image.
- Handle multiple input streams.
- Support different video input standards and output formats.
- Support buffering of video streams for subsequent processing (ultrasound 3D reconstruction etc.)

2.2 Design

During the design phase, the interaction of the video grabber component with existing and future IGSTK components was streamlined (Figure 2). The next step was designing a framework for the component based on the IGSTK state machine [1]. Figure 3 shows an illustration of the video grabber the state machine.
2.3 Implementation

As illustrated in Figure 1, platform specific classes of the video grabber components have to be implemented. Implementation starts with laying out the state machine that governs the functions of the video grabber component. State machines can help ensure that components are always in a known configuration. State machines contain a set of states, state inputs, and state transitions. IGSTK provides an `igstk::StateMachine` class that offers a set of public methods for programming, executing, and querying state machine logic. Figures 3 show the state machine diagrams for the `VideoGrabber` class. The class contains the following major states:

- **Idle**: Initial state.
- **GrabberReady**: Grabber ready to use.
- **GrabberActive**: Grabber activated. Allows `RequestGrabOneFrame()` calls.
- **Grabbing**: Grabbing video either to buffer, texture or both. A separate thread handles the updates.

In addition to these major states, transitional states exist that the grabber waits in until the requests are accomplished successfully. For example, the grabber makes a transition to `AttemptingToInitializeState` when a `RequestOpen()` method is invoked. Instead of `Set...()` methods that set parameters directly, the video grabber has `RequestSet...()` methods with a corresponding `AttemptingToSet...State` that can verify that the input value of the parameter is valid.
2.4 Results

A first version of the video grabber component was implemented using the QuickTime framework. This code is available as open source (a Berkley Software Distribution-like license) in the IGSTK sandbox along with an example application. The QuickTime code is based on a video grabber implementation from CustusX [6], a research and development platform for image-guided surgery. The first version of the IGSTK video grabber was only implemented for Macintosh OS X.

2.5 Future work

The current implementation of the video grabber uses a deprecated branch of the QuickTime framework in an attempt to provide functionality on both Windows and OS X. However this approach seems to produce some problems, as some of the deprecated code no longer functions correctly on the new Intel Macintosh computers, so a future version might have to use the newer OS X specific functions in Core Video and Core Image. A Windows implementation will probably have to rely on DirectX, while a Linux implementation may need to use Video4Linux.

In order to be integrated into the main IGSTK branch the video grabber component should have various tests to ensure that the code performs as expected. The grabbed video may have a small delay, and this delay is not the same as the delay of the tracking system. In order to get the correct image at the correct position a temporal calibration [7] is needed.
between the tracking and the grabbed video (as illustrated in figure 2). This is especially important for 3D ultrasound volume reconstructions [5].

3. EXAMPLE USE

The video grabber component will allow access to real-time data in the operating room. This data could be combined with preoperative data such as CT and MR images. The first use of the video grabber component will be to import 2D ultrasound data into IGSTK. The ultrasound probe has to be calibrated first [7] to find the transfer function between the tracked frame attached to the probe and the ultrasound scan plane. During surgery, the preoperative data is first registered to the patient reference frame, and then the ultrasound is imported into the same coordinate system. The ultrasound data may then be presented in different ways:

- A 2D presentation in a 2D view.
- A 2D presentation in a 3D view as shown in Figures 5 and 6.
- As a 3D reconstructed ultrasound volume.

All presentations could be combined with preoperative data in several possible ways. An example is a 2D slice through a reconstructed 3D ultrasound volume following a surgical tool, combined with a 3D preoperative volume. This allows the use of the most recent data during surgery while still showing the preoperative data. To allow for optimal use of the imported data, both a 3D ultrasound reconstruction and several visualization modes probably should be implemented. Both volume rendering and multiple volume visualization are modes that may enhance the user interface in addition to the 2D visualization modes that exist in IGSTK today.
3.1 Simple example

A simple example application was completed and submitted to the IGSTK sandbox. The IGSTK sandbox is a testing environment for newly implemented code that is not stable enough to be integrated into the main IGSTK branch [2]. The example code contains simple implementations of a spatial object and a representation object, both needed for visualization with IGSTK. This software is currently only running on the OS X operating system on a Macintosh computer (Macintosh, Apple, USA) (Figure 4a). The testing was performed on an abdominal phantom (Model 57, CIRS Inc., USA) (Figure 4b). Ultrasound video was obtained from a System FiVe ultrasound machine (GE Vingmed Ultrasound, Norway) (Figure 4c) with a Curved Linear Array probe with center frequency 3.5 MHz (Figure 4d). The video from the ultrasound scanner was converted by a Video-to-FireWire converter (DFG/1394-1e, The Imaging Source, Germany) (Figure 4e). Positions were acquired with an optical position tracker (Polaris Spectra, Northern Digital Inc., Canada) (Figure 4f). The optical position tracking device attached to the ultrasound probe comprises of 4 reflecting spheres (Figure 4d).

The example application is able to track the movement of an ultrasound probe and import and visualize the ultrasound image and position in real-time in a 3D scene (Figure 5).

Fig. 5. Image from the example application showing a tracked ultrasound probe with the grabbed image in an otherwise empty 3D scene.

3.2 Real world application.

An extended application simulating real-life use was also completed. This extended application uses the same setup as the previous example, but with a few additions. The IGSTK Video grabber is integrated into a new version of CustusX [6] (SINTEF, Norway), a research and development platform for image-guided Surgery. This new CustusX version is based partly on IGSTK. All the software ran on a desktop computer (Macintosh, Apple, USA). Preoperative data was acquired by scanning the phantom with a CT scanner (Sensation 64, Siemens, Germany) (Figure 4g). Skin fiducials, donut-shaped markers (15 mm diameter, 3mm thick, 4 mm hole), were glued to the phantom prior to the CT scanning.
These markers were used to register the 3D CT data to the physical phantom. The registration process was performed with a tracked pointer (Figure 4h). In addition to the CT volume, segmented objects from the CT volume were also imported into the visualization software (CustusX). The objects were segmented from the CT data with ITK Snap [8].

The real-life simulation is able to use both preoperative and intraoperative data. For this test, a CT image is used as preoperative data. This CT image is registered to the physical phantom, allowing the imported real-time ultrasound images to be positioned correctly with regard to the CT images of the phantom. This allows the use of real-time data together with preoperative data (Figure 6).

**4. CONCLUSIONS**

It is feasible to integrate real-time data into IGSTK with the purpose of providing more relevant and updated data during surgery. IGSTK, as an open source project allows researchers to extend it as needed for additional functionality, while supplying a structure to allow for development of robust code. With continued enhancements such as the video component described here, IGSTK may be a suitable toolkit for fast prototyping and development of safe and reliable image-guided surgery applications, including applications incorporating intraoperative imaging.

**ACKNOWLEDGMENTS**

Integrating support for real-time 2D medical image acquisition in the Image-Guided Surgery Toolkit (IGSTK) is a collaboration between SINTEF Health Research, Georgetown University, Kitware Inc., Arizona State University, and Atamai Inc. The toolkit as well as the VideoGrabber component and the simple example illustrating its use is freely available for download and can be used in research or commercial applications. More information can be found on the website at http://www.igstk.org.

The development of the VideoGrabber component is supported by the Research Council of Norway, through the FIFOS Programme Project 152831/530; the Ministry of Health and Social Affairs of Norway, through the National Centre of 3D Ultrasound in Surgery; and by SINTEF Health Research.
The Image-Guided Surgery Toolkit (IGSTK) is funded by NIBIB/NIH grant R01 EB007195 under project officer John Haller. Additional support was provided by U.S. Army grant W81XWH-04-1-007, administered by the Telemedicine and Advanced Technology Research Center (TATRC), Fort Detrick, Maryland.

We thank our other collaborators throughout the project, including Janne Beate Lervik Bakeng, and Arild Wollf from SINTEF Health Research, Medical Technology (and the National Centre for 3D Ultrasound in Surgery), Norway; Ziv Yaniv from Georgetown University, USA; Matt Turek from Kitware Inc., USA; Kevin Gary from Arizona State University, USA; and Nobuhiko Hata of Brigham and Women’s Hospital, USA.

REFERENCES