An Open-Source Solution for Interactive Acquisition, Processing and Transfer of Interventional Ultrasound Images

Release 0.10

Jonathan Boisvert[1], David Gobbi[1], Siddharth Vikal[1], Robert Rohling[2,5], Gabor Fichtinger[1] and Purang Abolmaesumi[1,3,4]

July 18, 2008

[1]School of Computing, Queen's University, Kingston, Canada
[2]Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada
[3]Department of Electrical and Computer Engineering, Queen's University, Kingston Canada
[4]Department of Surgery, Queen's University, Kingston, Canada
[5]Department of Mechanical Engineering, University of British Columbia, Vancouver, Canada

Abstract

Ultrasound has become a very important modality in image-guided therapy. At present, however, the collection, synchronization and transfer of ultrasonic images are more cumbersome than necessary. This paper presents a reusable solution to these problems. We propose a software package called SynchroGrab, which allows the collection of interventional ultrasound images as well as their synchronization with a stream of pose measurements. The software includes support for an open-interface ultrasound system, namely the Sonix RP, from Ultrasonix (Vancouver, Canada). Using an open-interface system like the Sonix RP allows customization of the imaging process and the capture of the ultrasound images directly from memory without the need for a frame-grabbing card. Pose measurement is currently performed with an Optotrak Certus by Northern Digital (Waterloo, Canada). However, extensibility was a primary goal in the design of this software, so the support of new devices can be achieved simply by subclassing the relevant base class. SynchroGrab also performs reconstruction of 3D ultrasound volumes from synchronized data streams. Moreover, the recorded images, volumes and tracking information are available for visualization or further processing either directly from the file system or from a network connection compliant with the OpenIGTLink protocol, which is supported by Slicer 3.

Latest version available at the Insight Journal [http://hdl.handle.net/1926/137]

Contents

1 Introduction

Over the years, ultrasound imaging has become a very important modality in the field of image-guided therapy (IGT). Its advantages are numerous and include very high availability of the equipment, innocuity for the patient and the operator, cost efficiency, and interactivity. However, ultrasound images are generally noisy and provide only limited information about a patient's anatomy. Thus, during image-guided therapies, ultrasound is typically used in combination with high-quality images acquired with computed tomography or magnetic resonance imaging prior to the intervention.

In addition to multi-modal imaging, IGT often relies on position sensors to track the position and orientation of different objects during interventions. Surgical instruments can hence be located in 3D, and displayed in the context of pre-operative images. Moreover, if calibration procedures are performed on a tracked ultrasound probe, then each pixel of the ultrasound images can be associated with a position in 3D space, and a 3D volume can be reconstructed from a sequence of 2D images.

For these technologies to be applied as part of an IGT procedure, the interventional ultrasound images, previously acquired images, and pose measurements must be imported into a common workspace. Once this information is regrouped, image visualization, analysis, registration and image fusion become possible. Unfortunately, commercial ultrasound systems do not generally provide such a common workspace, and the data transfer mechanisms that they do provide are usually not compatible with the real-time requirements of IGT.

Some surgical navigation systems tackle this problem by providing large, monolithic applications that perform all functions within a single integrated package. IGSonix (BrainLAB AG, Heimstetten, Germany), SonoNav (Louisville, CO, U.S.A.), and CustusX [6] are good examples of this practice. However, from a software engineering perspective, it would much more interesting to split some of these features into reusable components. On the other hand, open-source libraries such as IGSTK [4] offers some reusable components for IGT, but do not necessarily integrate well with existing systems.

In 2004, Boctor et al. [2] discussed the development of a module for Slicer 2 [7]. The module was intended to reconstruct volumes from tracked ultrasound images, and to make them available in Slicer 2 for further visualization and/or processing. This was a promising approach because Slicer is an open-source application that already offers a modular framework for advanced visualization and medical image processing. However, Slicer recently underwent a large re-engineering effort and the ultrasound module, which was intimately tied to the original Slicer 2 architecture, is no longer available for use.

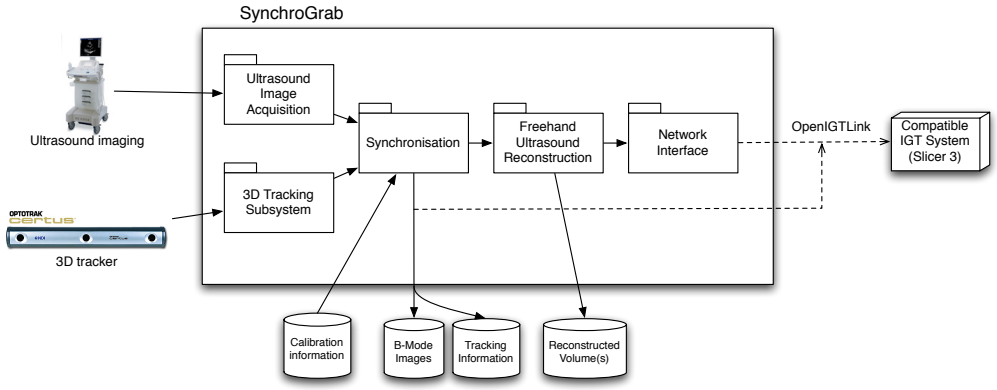

Figure 1: Information flow in SynchroGrab.

The recent proposal of OpenIGTLink [1], which allows the exchange of images and tracking information using a standardized network protocol, opens the door to a more flexible coupling between reusable software components in IGT. Moreover, with OpenIGTLink it is simple for different computers to manage different parts of the real-time image acquisition and interventional navigation. The former needs to comply with real-time constraints but typically does not require a large amount of computational resources, whereas the latter requires more computational resources and advanced computer graphics hardware but is more flexible with respect to timing constraints.

Another recent development is that open-interface ultrasound systems have become common. These allow more control over the types of ultrasound images collected and also offer direct access to the digital images rather than relying on digitization of an analog video stream.

In this context, we propose a new software package called SynchroGrab. This application fulfills the following requirements:

- Continuous acquisition of ultrasound images using an open-interface ultrasound system;
- Continuous acquisition of tracking information using a pose measurement system;
- Explicit synchronization of the two acquisition threads;
- Reconstruction of 3D ultrasound volumes using the synchronized data; and
- Transfer of the raw ultrasound images, the 3D volumes and the probe tracking information to an OpenIGTLink-compliant system.

The architecture of the proposed software is presented in the next section. Some of its components will be described in greater depth since they are likely to be reused in other applications. Section 3 then presents some experimental results obtained using SynchroGrab. These results demonstrate the most important features of the proposed software. Finally, future work and possible improvements to the architecture is discussed in Section 4.

2 SynchroGrab : Software Design

SynchroGrab was designed with several objectives in mind. The most important was to fulfill the requirements described in Section 1. However, the development of reusable software components was also judged to be very desirable. To achieve these objectives, the application was split into several logical modules with well-defined responsibilities. These modules, as well as their relationships, are presented in Figure 1.

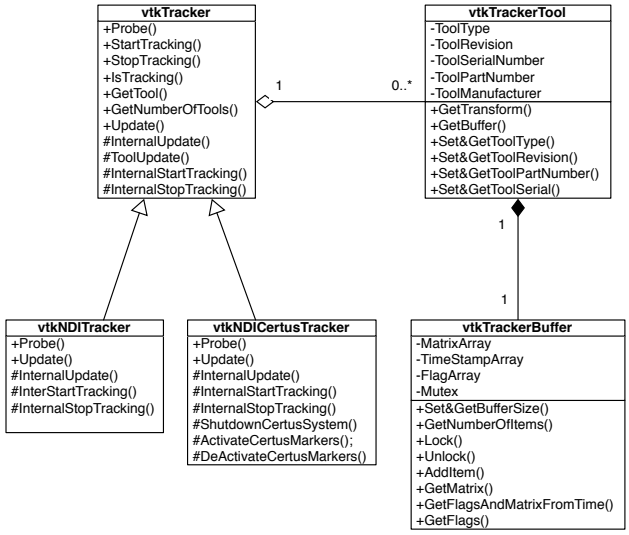

Figure 2: Class diagram of the 3D tracking subsystem.

The actual implementation of these modules corresponds to one or several C++ classes. Whenever possible, these classes were developed by subclassing the relevant VTK classes. This design decision facilitated the reutilization of existing VTK classes in SynchroGrab, and will also simplify the reutilization of our classes in other VTK-based applications.

2.1 3D Tracking

Figure 2 illustrates the most important C++ classes of our infrastructure, as well as their relationships. The base class `vtkTracker` is an open-source class from Atamai Inc. (London, ON, Canada) and provides an abstraction of the physical devices that generate rigid body tracking information. Two derived classes that represent concrete implementations of `vtkTracker` are `vtkNDITracker` and `vtkNDICertusTracker`. The former supports the AURORA and the POLARIS measurement systems from Northern Digital Inc. (Waterloo, ON, Canada), since both devices support a common API, while the latter supports the NDI Optotrak Certus, since the Certus has its own API. Other tracking systems can also be supported if concrete implementations of `vtkTracker` are written for them.

Once the correct concrete class is instantiated and communication with the tracking device is established via the `vtkTracker::StartTracking()` method, the objects belonging to the helper class `vtkTrackerTool` will provide information about the tools currently tracked. The tool poses, status information, and data timestamps are stored in a circular buffer to provide flexibility in synchronizing the poses with ultrasound images.

2.2 Ultrasound Image Acquisition

Most commercial ultrasound systems do not allow direct digital access to the captured images during the acquisition process. To alleviate this problem, most IGT systems support real-time ultrasound image acquisition by connecting a frame-grabbing card to an analog video output of the ultrasound system. This approach has many drawbacks. First, it is not possible to gain access to the raw RF signal or to pre-scan converted images. Second, a significant delay is created by the analog to digital and digital to analog conversions that

must be performed. Finally, image quality is likely to be decreased by such conversions.

An appealing alternative is the use of an ultrasound system that offers an open interface for data acquisition. By using the API of one of those systems, access can be gained to the ultrasound images directly from the system's memory without the need for any prior conversions. It also becomes possible to standardize the imaging parameters since those parameters can be established using the same API. The system chosen for this study is the Sonix RP from Ultrasonix (Vancouver, Canada) because it is the most widely used open-architecture ultrasound system. Other systems, such as Siemens' Axius Direct URI or Terason's Echo Ultrasound, also offer access to RF data and could be integrated in the future.

It was decided that we would start with the `vtkVideoSource` that is part of the VTK distribution, and provide support for the Sonix RP API by creating a derived class which we have called the `vtkSonixVideoSource`. By doing so, we have made it possible to swap out our class with any of the other concrete implementations of `vtkVideoSource`, which include the `vtkWin32VideoSource` class for Video-for-Windows supported frame grabbers, and the `vtkMILVideoSource` class for Matrox (Montreal, Canada) frame grabbers. The ability to swap out the video source is very important, since it will not restrict our open source software package to the proprietary Sonix RP interface.

When the `vtkSonixVideoSource` class is used, the internet address of the Sonix RP system to which it should connect (the default being the local computer) can be set. It is also possible to customize the requested image type with the methods `SetImagingMode()` and `SetAcquisitionDataType()`. Hence, one can request post-scan converted B-Mode images, pre-scan converted B-Mode images and even raw RF data. The following code excerpt illustrates how the class is typically used.

```
vtkSonixVideoSource *sonixGrabber = vtkSonixVideoSource::New();
sonixGrabber->SetSonixIP("127.0.0.1");
sonixGrabber->SetImagingMode(BMode);
sonixGrabber->SetAcquisitionDataType(udtBPost);
sonixGrabber->Record();
sonixGrabber->Stop();
sonixGrabber->Rewind();
// Perform image processing on the first image of the buffer here
sonixGrabber->Seek(1)
// Perform image processing on the second image of the buffer
// ...
sonixGrabber->ReleaseSystemResources();
sonixGrabber->Delete();
```

The method `SetImagingMode()` takes an integer as input but a C++ `enum` type enables the user to type clearer text values which are: `BMode`, `MMode`, `ColourMode`, `PwMode`, `RfMode`, `F4DMode`, `ElastoMode`. The method `SetAcquisitionDataType()` also takes an integer as input and also has an associated `enum` type for increased code readability. The supported acquisition data type include: `udtBPost`, `udtMPost`, `udtPWSpectrum`, `udtElastoOverlay`, `udtBPre`, `udtMPre`, `udtElastoPre`, `udtColorRF`, `udtPWRF`, `udtRF`, `udtScreen`, `udtBPost32`, `udtColorPost`, and `udtElastoCombined`.

2.3 Synchronization

Synchronization is an important issue since two separate acquisition threads must be managed; one for ultrasound image acquisition and the other for rigid body tracking. Since these two threads are unlikely to operate at the same frequency, the time stamp associated with a given image will not exactly match one rigid transformation recorded by the rigid body tracking thread.

An easy solution to this problem would be the consolidation of the rigid body tracking and the image acquisition into the same thread. However, this solution is less than optimal in several aspects. First, it is likely to reduce the overall frequency of the measurements. Second, it would result in less reusable code, since the acquisition thread would assume that the application has access to both a rigid body tracking system and an ultrasound imaging system.

In SynchroGrab, the synchronization is instead handled by a class known as `vtkTaggedImageFilter`, which selects the rigid body transformations measured immediately before an image as well as the one recorded immediately after. The selection of the rigid transformations considers a configurable acquisition lag between the ultrasound image data and the tracking data. Then, a spherical linear interpolation is performed to compute the rigid body transformation for the ultrasound probe at the time that the ultrasound image was acquired. Finally, a calibration matrix is applied to this transformation in order to obtain the transformation from the image coordinate system to the world coordinate system. The calibration parameters are computed using the method of Chen et al. [3].

2.4 Ultrasound Volume Reconstruction

The next step after synchronization is 3D volumetric reconstruction. Logically, there are two ways in which the reconstruction can be coupled with an IGT system. The first approach is to transfer the 2D images and their associated tracking information to the IGT system, e.g. over a network connection, and have the IGT system perform the reconstruction. The second approach is to perform the reconstruction outside of the IGT system and then transfer the resulting 3D volume to the IGT system either by writing it to disk or by transferring it over the network. The former approach has the advantage that the data can be displayed on the IGT system immediately as it is acquired, but full-rate data transfer might not be possible depending on the network speed. The latter approach performs the reconstruction on the system that acquires the data and does not suffer from a network bottleneck, and has the further advantage that one IGT system can be swapped out for another. Our chosen approach is a hybrid of the two: the 3D data can be reconstructed outside of the IGT system, but the IGT system may instead request the 2D data for visualization and/or reconstruction at the available network data rate. After all data are collected and the 3D reconstruction is completed, the 3D volume itself is transferred to the IGT system.

A large number of algorithms currently exist to perform volumetric reconstruction from tracked ultrasound images. Solberg et al. [8] recently presented a review of those algorithms. Any one of those methods could theoretically be supported by our application. However, the real-time requirements of our application made us opt for the method presented by Gobbi and Peters [5].

2.5 Network Interface

For the transfer of pose measurements, ultrasound images, and ultrasound volumes from SynchroGrab to the IGT system, we have chosen to use the OpenIGTLink protocol [1], which provides a simple, functional network interface for transferring images, pose measurements, and control information between components of an IGT system. The data is transferred at a user specified rate to a TCP/IP socket, which the OpenIGTLink module of Slicer 3 then reads and makes available within Slicer for uses such as visualization, registration and segmentation. It should be noted, however, that no control mechanisms are currently implemented to preclude flooding the receiver. Therefore, the user must set a transmission rate appropriate to the available network bandwidth.

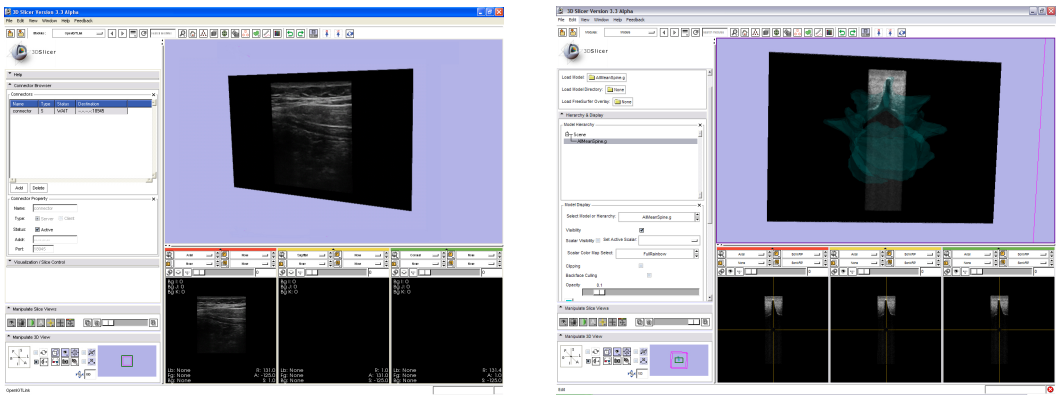

Figure 3: Left: Abdominal B-Mode ultrasound acquired with SynchroGrab and displayed by Slicer 3 in the position and orientation that corresponds to the ultrasound probe. Right: Lumbar phantom B-Mode ultrasound image collected using SynchroGrab and transmitted to Slicer 3 which also displayed a semi-transparent 3D model.

3 Capture Samples

A lumbar spine phantom and a volunteer's abdomen were imaged with SyncroGrab to demonstrate its most important features. The phantom was comprised of three artificial lumbar vertebrae purchased from Sawbones (Vashon, USA). These were embedded in a gel made of agar, gelatin, cellulose, glycerol, and water. To acquire the experimental data, we installed SynchroGrab on a Sonix RP system which was connected to an Optotrak Certus. The data were transmitted to a second computer (Windows XP, Intel Core2 2.4 GHz, 2 Gb RAM) running a copy of Slicer 3.3 with the OpenIGTLink module. Figure 3 provides samples of images received in real-time by Slicer.

4 Discussion and Conclusion

In this paper, we presented a software package called SynchroGrab, which addresses the problems of image acquisition, synchronization, reconstruction and transmission that are typical in image-guided therapy applications that rely on interventional ultrasound.

The proposed software first captures ultrasound images and then synchronizes them with a flow of tracking information. Then, it can reconstruct 3D ultrasound volumes, if desired. Finally, the resulting images, volumes and pose measurements can be saved to files or transmitted in real-time through a network interface to an image-guided therapy interface, such as Slicer 3. SynchroGrab supports the Sonix RP ultrasound system, so it becomes a simple matter to take full advantage of an open-interface ultrasound system in an IGT system.

The modular architecture of the proposed software ensures its extensibility. Therefore, adding support for new ultrasound systems or new tracking devices should be straightforward. More specifically, users of this system will be able to extend the framework in the following way: open the archive file containing the source code (provided with the paper), download the external libraries (from Northern Digital and Ultrasonix), build the source code using CMake, test the software with their own hardware, modify the source code to suit their particular needs, and finally share their modifications with the community.

Moreover, several C++ classes included in the SynchroGrab source code should be useful for other research teams, and we designed them to be easily reusable. Among these classes, we must mention:

`vtkSonixVideoSource` that allows one to integrate real-time ultrasound capture in a VTK pipeline, `vtkTaggedImageFilter` which synchronizes a flow of images to a flow of tracking information, and `vtkNDICertusTracker` which supports the Optotrak Certus in a highly extensible object-oriented framework.

One of the limitations of the current implementation is the fact that network communication is presently restricted to a single direction (one-way communication). In other words, SynchroGrab can send information to a compatible IGT system, but do not support incoming communications at the moment. Therefore, imaging parameters, such as frame rate or image depth, must be supplied to SynchroGrab upon launch and cannot be changed remotely. We plan to correct this situation by adding a series of commands that can be triggered remotely. Furthermore, we hope to develop a custom Slicer 3 module that will support those commands. Another notable improvement would be additional support for the control of 3D probes. Finally, additional support for new ultrasound systems and new tracking devices will also be considered. Third-party contributions will thus be welcomed.

5 Acknowledgement

The authors wish to thank the Natural Sciences and Engineering Research Council (NSERC), the Canadian Institutes of Health Research (CIHR), and the National Alliance in Medical Image Computing (NIH 5U54EB005149-03) for funding this project.

References

[1] *OpenIGT Link Protocol*. http://www.na-mic.org/Wiki/index.php/OpenIGTLink/Protocol. 1, 2.5

[2] E. M. Boctor, A. Viswanathan, S. Pieper, M. A. Choti, R. H. Taylor, R. Kikinis, and G. Fichtinger. CISUS: an integrated 3D ultrasound system for IGT using a modular tracking API. In R. L. Galloway, Jr., editor, *Medical Imaging 2004: Visualization, Image-Guided Procedures, and Display. Proceedings of the SPIE*, volume 5367, pages 247–256, May 2004. 1

[3] T.K. Chen, A.D. Thurston, M.H. Moghari, R.E. Ellis, and P. Abolmaesumi. A real-time ultrasound calibration system with automatic accuracy control and incorporation of ultrasound section thickness. In *Proceedings of SPIE Medical Imaging*, 2008. 2.3

[4] 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

[5] D.G. Gobbi and T.M. Peters. Interactive intra-operative 3D ultrasound reconstruction and visualization. In *Proc. MICCAI (Lecture Notes in Computer Science Vol.2489)*, pages 156 – 63, 2002. 2.4

[6] T. Lango, G. A. Tangen, R. Marvik, B. Ystgaard, Y. Yavuz, J. H. Kaspersen, O. V. Solberg, and T. A. N. Hernes. Navigation in laparoscopy–prototype research platform for improved image-guided surgery. *Minim Invasive Ther Allied Technol*, 17(1):17–33, 2008. 1

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

[8] O.V. Solberg, F. Lindseth, H. Torp, R.E. Blake, and T.A.N. Hernes. Freehand 3D ultrasound reconstruction algorithms-a review. *Ultrasound in Medicine & Biology*, 33(7):991 – 1009, 2007. 2.4