PLUS: Open-Source Toolkit for Ultrasound-Guided Intervention Systems

Article *in* IEEE transactions on bio-medical engineering · May 2014

DOI: 10.1109/TBME.2014.2322864 · Source: PubMed

6 authors, including:

Andras Lasso
Queen's University
84 PUBLICATIONS **394** CITATIONS
SEE PROFILE

Tamas Heffter
8 PUBLICATIONS **96** CITATIONS
SEE PROFILE

Csaba Pinter
Queen's University
19 PUBLICATIONS **137** CITATIONS
SEE PROFILE

Gabor Fichtinger
Queen's University
383 PUBLICATIONS **5,024** CITATIONS
SEE PROFILE

PLUS: open-source toolkit for ultrasound-guided intervention systems

Andras Lasso, *Member, IEEE,* Tamas Heffter, Adam Rankin, Csaba Pinter, Tamas Ungi, and Gabor Fichtinger, *Senior Member, IEEE*

***Abstract*—A variety of advanced image analysis methods have been under development for ultrasound-guided interventions. Unfortunately, the transition from an image analysis algorithm to clinical feasibility trials as part of an intervention system requires integration of many components, such as imaging and tracking devices, data processing algorithms, and visualization software. The objective of our work is to provide a freely available open-source software platform - PLUS: Public software Library for Ultrasound - to facilitate rapid prototyping of ultrasound-guided intervention systems for translational clinical research. PLUS provides a variety of methods for interventional tool pose and ultrasound image acquisition from a wide range of tracking and imaging devices, spatial and temporal calibration, volume reconstruction, simulated image generation, and recording and live streaming of the acquired data. This paper introduces PLUS, explains its functionality and architecture, and presents typical uses and performance in ultrasound-guided intervention systems. PLUS fulfills the essential requirements for the development of ultrasound-guided intervention systems and it aspires to become a widely used translational research prototyping platform. PLUS is freely available as open source under BSD license, the code and documentation are available at http://www.plustoolkit.org.**

***Index Terms*—tracked ultrasound, tool navigation, open-source, spatial calibration, temporal calibration, volume reconstruction, live image streaming**

I. Introduction

MEDICAL interventions take great advantage of ultrasound (US) guidance, as US is a safe, real-time, low-cost, and widely accessible imaging modality. US-guidance is already the clinical standard in various forms of injections, biopsies, ablations, cannulations, and other procedures ([1], [2], [3], [4]). Some of these interventions are simply guided by free-hand manipulation of the US transducer and the interventional tool. However, more challenging procedures cannot be performed solely relying on US images, or demand extraordinary manual skills. In such difficult procedures, US imaging can be equipped with spatial tracking to enable further visualization and navigation techniques.

Difficulties in US guidance are poor target visibility or difficult manual coordination. Spatial tracking may help overcome both of these. When the target of the needle insertion is deep under the skin, or has similar echogenicity as the surrounding tissues, it is difficult to recognize the target in US images. Target visibility may be enhanced by the fusion of real-time US images with pre-operative computed tomography (CT) or magnetic resonance (MR) images ([5], [6], [7]). Such image fusion has already been developed for some clinical applications with the help of US tracking, and it has the potential to transform the way many other radiological interventions are performed. Also, US imaging has a limited field of view. US may fail to show the necessary anatomical context for certain procedures, e.g., identification of a spinal segment is difficult from just one US image. Tracked US can be extended by stitching together many US image slices and reconstructing them in a larger 3D image volume [8].

The second major difficulty with US guidance is the coordination of hand-held tools, i.e., the US transducer and the needle. Although the needle entry point and the target point may be visible in a single US image, the transducer has to be moved away when starting the needle insertion to make space for the needle. Therefore, the insertion has to be started blindly and the needle path may have to be corrected later, when the tool appears in the image. This targeting problem is not entirely solved by mechanical needle guides attached to the US transducer. They limit the relative position and orientation of the needle and the transducer, which may force the operator to choose between optimal needle path and best image quality. A US-integrated tracking system, on the other hand, measures the pose (i.e., position and orientation) of the US transducer, interventional tools, and the patient. It enables tool pose visualization at any location, not just within the field of view of the imaging device, and without limiting the freedom of motion of the operator.

Medical device manufacturers realized the great potential of tracked US and several vendors now offer products that utilize this technique. However, most commercially available systems provide only basic features, such as real-time display of tool position, point marking, point-based registration, and volume reconstruction, reslicing, and fusion. Introduction of more advanced guidance techniques into products, such as automatic image or atlas-based registration, image-based tracking, robotic assistance, real-time motion compensation, real-time multi-modality image and data fusion ([9], [10], [11], [12], [13], [7], [14], [15]) need extensive explorative and development work.

Research and development of tracked – also known as navigated – US-guided intervention systems require advanced engineering infrastructure, which has not been available in the public domain, and single project-based solutions have not proven to be suitable as reusable platforms.

Obtaining real-time image and pose data from the commer-

Andras Lasso, Tamas Heffter, Csaba Pinter, Adam Rankin, and Gabor Fichtinger are with the Laboratory for Percutaneous Surgery, School of Computing, Queen's University, Kingston, ON, Canada, e-mail: {lasso, heffter, pinter, rankin, gabor}@cs.queensu.ca

cial ultrasound systems for research purposes is a challenging task. Many vendors require the completion of lengthy legal procedures to allow direct access to their devices. A few vendors offer open interfaces for data acquisition and control of their systems. These open interfaces enable advanced research work without administrative burden or custom hardware development. However, these open interfaces are all vendor-specific, therefore developers have to invest significant amount of time into making their software work with each device. As a workaround, on closed ultrasound systems live image data may be acquired by connecting a framegrabber on the video output of the device; however, the digitized image quality is often not optimal due to noise, limited resolution, and various annotations overlaid on the displayed image, and important supplementary information – such as imaging depth or zoom factor changes, image freeze status – are not available. In certain cases, tracker-free, image-based methods can be applied for obtaining pose information ([16]); however, these methods are so limited (allow tracking only of the ultrasound probe, suffer from drifting, do not offer absolute position measurement) that they are out of scope of this paper.

Boisvert, *et al.* developed the open-source SynchroGrab software library [17] for the acquisition of tracked ultrasound data using a unified interface, with a number of different tracking and imaging devices. SynchroGrab also contained algorithms for reconstruction of a volumetric image from the acquired US image slices. SynchroGrab was based on the open-source Visualization Toolkit (VTK, [18]) and worked with a couple of ultrasound and tracking devices. Later, Pace *et al.* [19] extended the library with electrocardiogram gating to allow reconstruction of time sequences of volumetric images. SynchroGrab was a valuable contribution to this field and served as a good example with its architecture, utilization of open communication protocols and toolkits. However, the library lacked some essential functions, such as limitation to supporting only one tracker and imaging device at a time, lack of spatial and temporal calibration, recording capability, diagnostic and configuration tools, documentation, samples, and tests.

The Image-Guided Surgery Toolkit (IGSTK, [20]) is a generic open-source framework for image-guided surgery applications. It supports pose tracking using a number of hardware devices, contains some registration functions, and provides a robust application infrastructure. There were attempts to add US image acquisition and spatial calibration features to IGSTK ([21]), but these functionalities have not become part of the toolkit. Using or extending IGSTK for tracked US implementation is also complicated by the toolkit's unique architecture, which extensively uses state machines in every part of the software. Application of state machines can improve flexibility and robustness of the software, but using them to specify all behaviors in a large software application requires lots of experience and often not ideal for implementation of computational algorithms: as core IGSTK developers explain in [21] why they have not implemented their algorithm inside their toolkit: "...the architecture is cumbersome for algorithmic development. As a consequence we have implemented our algorithms as an external library".

Stradwin is a software application developed by Treece *et al.* [22] at the University of Cambridge, UK, for freehand 3D ultrasound calibration, acquisition, measurement, and visualisation. While Stradwin has many useful features, the software is not designed for generic interventional tool guidance and the software's source code is not publicly available.

The Medical UltraSound Imaging and Intervention Collaboration (MUSiiC) research lab developed a toolkit [23] for acquiring and processing tracked ultrasound data. The reported characteristics of the toolkit are very promising, but the toolkit has yet to be released to the research community as an open-source package.

Recently an US imaging extension has been added to the open-source Medical Imaging Interaction Toolkit (MITK): MITK-US [24]. This new component, combined with existing tracking and visualization components in the toolkit can be used for developing simple US guidance systems, i.e., that only require showing tracked B-mode images and tools, and do not need volume reconstruction or RF-mode imaging. MITK-US currently does not include any US image processing or calibration methods.

There are a couple of other frameworks for development of research systems, which support acquisition of real-time tracking and image data. Examples include the Computer-Integrated Surgical Systems and Technology (CISST) libraries developed at Johns Hopkins University [25], the OpenTracker library[26], and the Virtual Reality Peripheral Network (VRPN) library [27]. These toolkits are open-source, have useful components, such as hardware device interfaces and data collection and processing infrastructure, but focused more on robotic systems (CISST) and augmented reality applications (OpenTracker, VRPN) and so lack several features (limited support for imaging, no interface to medical US imaging systems, no US image calibration and processing algorithms, etc.) that are essential for US-guided interventions.

In this paper, we present PLUS: Public software Library for UltraSound. The purpose of PLUS is to provide a freely available open-source toolkit to facilitate rapid development of US-guided intervention systems for translational clinical research. The PLUS toolkit offers access to various US imaging and pose tracking tools using a single, hardware-independent interface and includes software, hardware, documentation, and know-how for commonly needed calibration and data processing, visualization, and transfer operations.

The main methodological contributions of this paper can be summarized as follows: (1) System architecture that allows decoupling of imaging and pose tracking hardware devices from algorithms and end-user software applications; (2) A comprehensive and easy to implement set of definitions for describing spatial and imaging data, as well as auxiliary application-dependent configuration descriptors; (3) Extensive set of ultrasound image manipulation utilities implemented in an integrated framework.

II. Methods

In this section we first identify the minimum requirements for a toolkit that offers a solution for rapid development of US-guided intervention systems for translational clinical research,

then describe the architectural, design, and implementation work completed to fulfill these needs.

A. Requirements

The requirements for an US-guided intervention system are quite different for research systems and commercial products. We set the requirements to fulfill the needs of rapid system development for translational clinical research.

1) Functional requirements: The toolkit should offer the following functionalities: acquisition of US image and pose tracking data; switching between various data acquisition devices without software modification; persistent storage of the acquired data in files; continuous real-time transfer of the acquired data to other software or systems; spatial and temporal calibration of tracked US image data; reconstruction of volumetric images from tracked US image slices; include tools for diagnostics and troubleshooting; contain end-user applications that implement basic functionalities with a convenient user interface and also serve as application examples.

2) Non-functional requirements: Openness: As the goal is to provide a freely available framework, all custom software and hardware components should have a license that enables any use, modification, and distribution without limitations. The Berkeley Software Distribution (BSD) license allows use in a commercial product and does not enforce anybody to share custom modifications or enhancements with others.We require the toolkit – including source code, documentation, tutorials, examples, tests, CAD drawings, and all software libraries that it relies on – to be compatible with the BSD license. Well-established, open, royalty-free standards should be used whenever they are available to minimize reimplementation workload and maximize interoperability.

Extensibility: The toolkit is intended to offer core functionalities, which are extended by the researchers when investigating particular problems. Extension of the toolkit with new functionalities should be simple and contributing the new functionalities to the community should be possible to do with minimal overhead.

Maintainability: It is essential to preserve existing functionality and quality of the toolkit while functionalities are changed or added. It should be possible to detect and resolve regressions in the software with minimal effort.

We do not require high degree of *robustness*. Reducing probability of failures to negligible levels is enforced by regulatory agencies for commercial medical products. However, the enormous efforts that are needed to achieve and prove the robustness of a clinical-grade software usually do not pay off in a research setting, where software failures tend to be more tolerable. *Efficiency* is not listed as a requirement either, as performance-optimized software is usually more complex and difficult to change, which is not desirable in the research phase of a project. Therefore, users of research systems may have to tolerate slightly longer computation times and slower feedbacks, compared to what they expect from single-purpose highly optimized commercial systems.

B. Data representation

Defining common notations and formats for basic data structures is required for seamless interoperability between different groups that use, maintain, and extend the tracked ultrasound system. The basic information types that have to be represented are image, pose, and time.

1) Image: The PLUS toolkit uses the image representation defined in its foundation libraries (ITK[28] and VTK[18]). However, there is an additional specific property – image orientation – that has to be specified for each US image slice. In PLUS, image orientation refers to the spatial relationship between the image axes and the transducer principal axes (marked/unmarked side; near/far from the transducer).

The DICOM standard (Part 3: C.8.24.2.1.2) describes a way to specify image orientation by a transformation between the image and the transducer frame of reference. However, the transformation can be determined only if the system is spatially calibrated and the transducer surface centerpoint position in the image is known.

2) Pose: The pose of the acquired image slices, tools, and other objects are defined by specifying a 3D Cartesian coordinate system (a.k.a. reference frame) for each object and transformations between them. The transformation is assumed to be rigid and each transformation is represented by 4x4 homogeneous transformation matrix. Each coordinate system is defined by its name, origin position, axis directions, and scaling unit. These definitions must be completed for all coordinate systems and made known to the developers of the system to avoid any chance of misunderstandings.

One of the most frequently needed operations in image-guided intervention systems is to compute the transformation between two arbitrary reference frames. Although this operation is very simple in theory, the implementation can be quite complex and error-prone in practice, mainly due to the large number of transformations and lack of clear naming and specification of transformations. The number of potentially needed transformations is high even for a simple case: a tracker with 3 tracked objects typically uses 7 coordinate systems (3 markers, 3 objects that the markers are attached to, 1 tracker), which leads to 42 different transformations.

A commonly used technique is to store all transformations as edges of a graph, where each vertex of the graph corresponds to a coordinate system and compute the transformations automatically (see e.g., IGSTK, [20]). The automatic computation is straightforward: first the path (list of transformations) between the two coordinate systems (vertices) is searched in the graph, then the corresponding transformations are chained in the correct order and inverted as needed. This approach is implemented in PLUS: all transformations and their inverses are stored in one system-wide directed acyclic graph, where each vertex is identified by the coordinate system name. Transformation between any two coordinate systems is computed by collecting all the transformations using a breadth-first search then multiplying the corresponding transformation matrices. Having one single repository and a consistent naming of coordinate systems allow unambiguous definition of transformation names by a simple character string constructed from the names of the source and destination coordinate systems.

3) Time: It is often needed to assign timestamps to data items. In PLUS, *system time* is used internally for all timestamping. System time is measured by high-resolution timer in the computer and is defined to be 0 sec on application start. Hardware devices are assumed to assign timestamps to their acquired data in their *local time*. Therefore, these timestamps are converted to system time after acquisition. PLUS is also aware of *universal time*, which is the current local time in Coordinated Universal Time (UTC) and used when communicating with other software that require absolute timestamps.

C. Data acquisition

For prototyping of US-guided intervention systems it is desirable to make the implementation independent from underlying hardware as much as possible. This allows carrying out the same procedural workflow with any hardware device, without making any software change. Decoupling of the hardware-specific parts from the rest of the system also enables performance comparison of different hardware devices and reduces the amount of work for implementing interfaces to new devices.

Often data has to be collected from several hardware devices, such as multiple trackers or multiple imaging devices at the same time, which are then synchronized, processed, and transferred in various ways. This kind of data processing is most commonly implemented using a data flow pipeline architecture. For our case the pipeline has to be buffered (as temporal synchronization requires access to at least the last few seconds of acquired data), multi-threaded (as data is acquired from several hardware devices simultaneously), and has to support processing of live streaming data. Unfortunately, the pipeline implementations that were available in the toolkits that PLUS already relies on (VTK and ITK) do not fulfill all these requirements, therefore a custom data collection and processing pipeline had to be implemented.

In the pipeline, each hardware device or data processing algorithm is represented by a *device*. Each device may provide data through *output channel(s)*. A device can generate data internally (e.g., from data that it collects from a hardware device) and/or use data that it receives from another device's output channel. A *channel* transfers a bundle of data streams: for each time point it stores a single video frame and/or any number of attributes, such as transforms or status information. Each physical hardware device has a corresponding device in the pipeline. In addition, there are *virtual devices* that perform various operations on their input channel(s), such as fusion of multiple channels, disk storage, volume reconstruction, or simulated data generation. For optimal performance, when image data is transferred from one device to the other, image frames are actually not copied, but only a reference to the channel and the frame is passed. A channel keeps a predefined number of most recent data items in a circular buffer to allow temporal interpolation and avoid data loss due to delayed processing.

The pipeline can be used for handling dynamic changes in the geometry of the acquired US image. For example, a common use case is to allow the clinician to change the US imaging depth during the procedure. Changing the imaging depth typically changes the image size, and therefore the spatial calibration has to be updated dynamically, synchronized with the change in the image contents. The solution to this in PLUS is to create a separate channel for each supported imaging depth and associate each channel with the corresponding calibration information. Optionally a *switcher* virtual device can be used to redirect the multiple channels to one single output channel. This setup allows straightforward, static definition, calibration, and testing of each supported imaging depth. The method also guarantees that each image frame is always associated with correct spatial calibration information, so no invalid data is acquired during the transient time during depth changes. The same approach is used for supporting dynamic switching between data acquisition from different transducers.

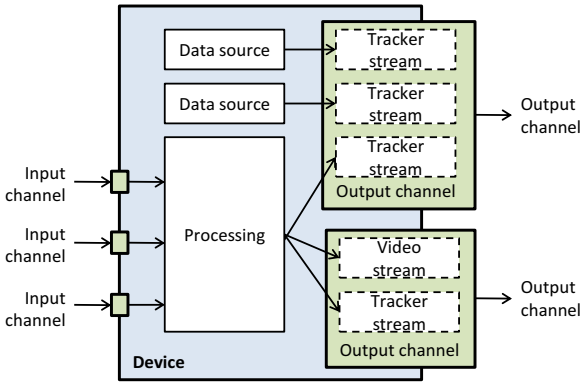

Figure 1. Example of a pipeline for simultaneous acquisition of tracked B-mode and RF image data. B-mode data is used for live display, while RF data is saved to disk.

US and tracking data are typically acquired by separate devices, therefore fusion of these data streams is necessary. As image frames and object poses are acquired at distinct time points, they have to be resampled before data fusion. Image data is more complex to interpolate and it has larger size, therefore in PLUS only the pose information is resampled, at each time point when an image frame is acquired. The position part of the pose is interpolated with weighted averaging, the orientation part is computed by spherical linear interpolation. The fusion of the data is implemented in the *virtual mixer* device.

Commonly, PLUS acquires brightness mode (B-mode) US images to guide interventions. In B-mode images pixel brightness represents the strengths of ultrasound echo from the corresponding spatial position. PLUS can also retrieve the unprocessed high-frequency (radio-frequency or RF) signal from certain US imaging devices, which contains more information than B-mode images, but cannot be readily displayed as an image. A basic processing algorithm is implemented in PLUS to convert the RF data to B-mode ultrasound images, in order to allow monitoring of the contents and quality of the RF data during acquisition. The processing of the RF data starts with brightness conversion, which computes the image pixel intensity values by envelope detection and log compression for

each scanline. Then scan conversion is performed by pasting the scanlines into the image slice. Scan conversion is supported for both linear and curvilinear transducer geometry.

The minimum dislocation and maximum acceptable speed parameters can be specified in PLUS for gating the data collection. Minimum speed can reduce the amount of acquired data when the imaging probe moves slowly. Maximum allowed speed can limit the amount of spatial error that is caused by temporal misalignment when moving the US probe quickly.

All the hardware configuration data – definitions of which hardware devices are used for acquiring the tracking and US image data, acquisition rates, connection settings, etc. – are specified in a configuration file, so that changes can be applied without modifying or rebuilding the software.

D. Temporal calibration

Finding the corresponding pose data for each image requires accurate timestamping for each item. US-guided intervention systems typically consist of three loosely coupled subsystems: pose tracker, US scanner, and data collector computer. Each hardware device provides timestamps by using its own hardware clock, which is not synchronized to the clock of the other devices. Some devices do not even provide a timestamp, but the receiving data collector computer records the acquisition time. Therefore, a temporal calibration method is needed, which can correlate the timestamps provided by the various data sources.

The temporal calibration method in PLUS assumes that data sources attach timestamps on the acquired data with an unknown but constant time offset. The output of the temporal calibration is the time offset between the different data sources that leads to optimal correlation of the pose changes. The input of the method is a single 10-second recording of tracking and imaging data while moving the transducer with a continuous quasi-periodic pattern (e.g., moving up and down by hand). The first step of the processing is to extract a 1D position signal acquired data. 1D signal from tracking data is computed by projecting the 3D position to the principal axis of the motion. Position signal from US image data is computed by placing a static planar object in the field of view and extract the position of the line from the image. The same method can be used for synchronization between tracker/tracker, tracker/imaging device, and imaging device/imaging device. Visual inspection of the aligned signals provides useful diagnostic information about the root cause of temporal misalignments, such as large occasional delays or varying acquisition rate.

If the data acquisition device does not provide timestamps, then the data collector software has to timestamp the data item when it is received. Timestamping on a personal computer is generally feasible with about 1ms precision when using multimedia timers. However, due to the non-real-time nature of general-purpose operating systems, slight variations in the timing may occur due to hardware interrupts and varying processor load. Also, some hardware devices transfer the data through Ethernet network, in which case there may be an additional unpredictable delay before the data collector receives the data.

If the acquisition hardware device acquires data items at regular time intervals and reports missing data items, then filtering that ensures regular time intervals between the generated timestamps and remove items with unreliable timestamps. Previously, a recursive filtering scheme was implemented in SynchroGrab [17]. However, we found it difficult to find a filter parameter that efficiently reduces the variance in the timestamp differences and at the same time always remains stable (in some cases the filter produced divergent oscillation). In PLUS we implemented a method in that a line is fitted using linear regression to the last N pairs of item indexes (independent variable) and unfiltered timestamps (measured variable), and then the filtered timestamp is retrieved for the frame index according to the fitted line. Compared to the recursive filter, this method is more controllable, as it never diverges or oscillates, and more predictable, as we can define exactly how long any sample may influence the filter output.

E. Spatial calibration

The goal of spatial calibration is to determine the transformation between the coordinate systems of an object (e.g., image slice, calibration phantom, stylus) and a marker that is rigidly attached to that object. A typical setup includes a tracked US transducer, tool (such as stylus or needle), and patient-attached reference sensor, as shown in Fig. 2. Three different calibration methods are implemented in PLUS.

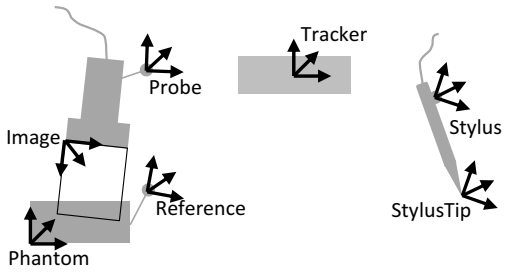

Figure 2. Tools and associated coordinate systems in a typical tracked US system calibration setup

Pivot calibration is used for computing the translation between the tip of a pointer tool (a.k.a. stylus) and the marker that is attached to the tool. Usually the marker's coordinate system is named as Stylus, the coordinate system that has its origin at the tooltip is named as StylusTip.

Landmark registration is used for computing the transformation between an object and the attached marker, in cases when there are known positions on the object that can be touched with a stylus. E.g., this method is used to determine the transformation between the calibration phantom's coordinate system (Phantom) and the attached marker's coordinate system (Reference).

The most challenging calibration is the determination of transformation between the coordinate system of the image (Image) and the marker that is attached to the US transducer (Probe). A review of potential methods is given in [29]. In PLUS, a multiple N-shaped fiducial based calibration method [30], [31] is implemented, as it is accurate, easy-to-perform,

fully automatic, and the required hardware parts are easy to obtain. The position and size of the N-shaped fiducials are configured in a file, so that the calibration phantom can be easily modified without requiring any software change. To ensure reproducibility of the calibration method, detailed description of calibration phantom, its 3D-printing-ready CAD model, and assembly instructions are provided in the PLUS project's documentation.

F. Volume reconstruction

US volume reconstruction methods construct a 3D Cartesian volume from a set of 2D US frames that are sweeping across a region. The volume may be used for many purposes, including 3D visualization, multi-planar reconstruction, and image-based registration.

The basic volume reconstruction method in PLUS is based on the work of [32] and Boisvert *et al.* [17]. The first step of the procedure is insertion of 2D image slices into a 3D volume. This is implemented by iterating through each pixel of the rectangular or fan-shaped region of the slice and inserting the pixel value into the corresponding volume voxel ("nearest-neighbor interpolation" option) or distributing it in the closest 8 volume voxels ("linear interpolation" option). Linear interpolation preserves more information from the original 2D slice, however it requires a higher-resolution 3D volume and more computation time. The voxel value can be determined by simply using the latest coinciding pixel value or as a weighted average of all coinciding pixels ("compounding" option). Slice insertion can be typically performed at the rate the images are acquired, therefore individual image slices does not have to be stored and the reconstructed volume is readily available at the end of the acquisition.

This basic algorithm was improved with additional features in PLUS. Options added for computing voxel values as the minimum or maximum of the coinciding slice pixel and volume voxel value ("minimum" or "maximum" options), which is useful for removing acoustic shadows or other artifacts by multiple image sweeps of the same region. There is also an averaging ("mean" option) that reduces the random noise in the reconstructed image.

The spacing between the acquired image slices and their orientation may vary, while the reconstructed volume has uniform spacing along each axis. If the reconstructed volume spacing is set to match the largest gaps between the acquired slices, then the volumetric image resolution will be low. If the resolution of the output volume is set to a higher value then a simple pasting of slices into the volume results in unfilled regions, "holes" in the reconstructed volume. To avoid holes in the output volume a low resolution volume can be reconstructed, with a spacing that corresponds to the largest spacing between the 2D image slices, but this leads to loss of details in the reconstructed volume. A hole-filling method is implemented in PLUS that enables reconstruction of a high resolution volume and removes the holes by interpolating from nearby voxel values. The hole-filling method computes weighted averaging of nearby known voxels with a varying size spherical Gaussian kernel, or with an elliptical kernel. The orientation and size of the elliptical kernel is computed automatically for each voxel to minimize blurring in the filled volume regions.

The volume reconstruction algorithm is integrated into a virtual device that can be connected to any output channel that contains tracked image data. The generated volume is immediately accessible after frames have been added, which allows live visualization or saving to disk while collecting frames.

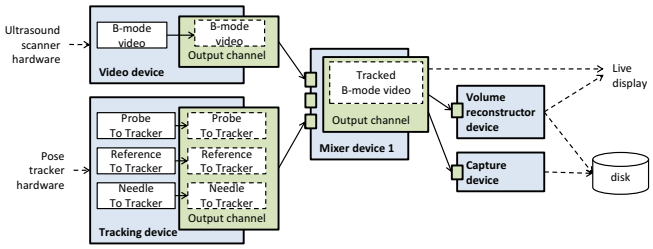

Figure 3. Example of a pipeline for volume reconstruction.

G. Ultrasound simulation

Generation of synthetic US images is useful for several purposes, such as user training and software application and algorithm testing and optimization, therefore this functionality was added to the toolkit ([33]). The most important property of the simulator is that any number of moving, intersecting objects can be simulated. Each object is defined by its acoustic material properties and a surface mesh, as interventional tools are already specified by surface meshes and segmented anatomical objects can be also represented well using surface meshes. Position and orientation of objects can be obtained in real-time from any tracking device or from pre-recorded files. Inidividual scanlines are computed using a simple ultrasound physics based model, which includes attenuation, absorption, diffuse and specular surface reflection, and speckle (using Perlin noise). Multiple reflections, refraction, speed of sound, and beamwidth are excluded from the model. Both linear and curvilinear transducer geometry is supported. With minor modification in the device set configuration file image acquisition can be switched to use a real ultrasound device. The algorithm is implemented as a virtual device, a typical setup is shown in Fig. 4.

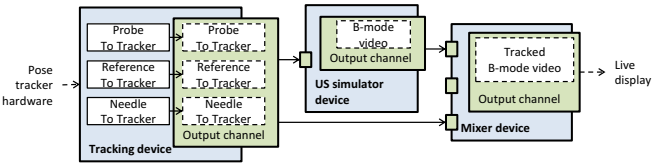

Figure 4. Example of a pipeline for generating simulated ultrasound data for a tracked probe and needle.

H. Capture to disk

Recording of data to disk is implemented as a virtual device, which can store a predefined number of frames in memory and

write to file when the buffer is filled. This enables recording of bursts of high-frame-rate data without losing any frames (by writing to memory) and also saving long, continuous acquisitions without running out of memory (by writing to disk). A typical pipeline that includes a capture device is shown in Fig. 5.

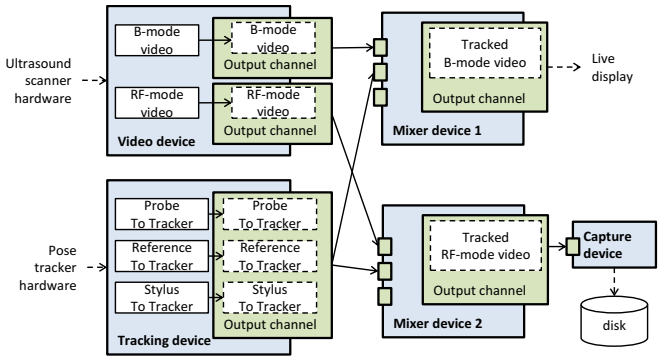

Figure 5. Example of a pipeline for simultaneous acquisition of tracked B-mode and RF image data. B-mode data is used for live display, while RF data is saved to disk.

I. Live data streaming

US-guided intervention systems often have to be implemented in a loosely integrated, heterogeneous environment. Data collection, processing, and visualization may need to be performed in different processes, sometimes on different computers. To fulfill these needs, PLUS provides live streaming output and it also accepts live streaming data input through the OpenIGTLink protocol [34]. OpenIGTLink is an open, very simple, light-weight, TCP/IP based communication protocol that is specifically developed for live data transfer for image-guided therapy applications. Several hardware devices (e.g., position trackers, imaging devices, robotic systems) and software applications (e.g., 3D Slicer, MeVisLab) supports live data sending and receiving using this protocol.

PLUS includes a standalone server application (PlusServer) that can acquire data from multiple hardware devices and/or through OpenIGTLink, fuses all data, then sends them to any number of connected clients. Clients can subscribe to a subset or all of the available data. PlusServer can be connected to another instance of PlusServer or any other OpenIGTLink-compatible software application, which makes possible to set up a distributed data acquisition, processing, and visualization pipeline across multiple processes, running on multiple networked computers.

PlusServer connected clients not just receive data but can also control PlusServer by sending commands to devices through using OpenIGTLink. Commands already exist for starting and stopping recording, performing volume reconstruction, and for requesting the transfer of reconstructed images. Developers can easily implement new custom commands in the server as needed.

J. Implementation

1) System architecture: The PLUS software is divided into two parts: library and applications (Fig. 6). The PLUS library contains the implementation of all the algorithms, code for data collection, interfacing with hardware devices, tests, examples, and basic common classes. The PLUS applications include an end-user application (fCal) for performing the calibration steps of a free-hand US system, a standalone application for volume reconstruction, a server that can forward acquired data to multiple clients using OpenIGTLink, and a number of diagnostic and test tools. PLUS relies on other free, open-source libraries that are commonly used for the implementation of medical image computing systems: ITK, VTK, and OpenIGTLink. PLUS applications use the Qt toolkit for graphical user interface (GUI) implementation. PLUS also uses device drivers and software development kits provided by the device manufacturer for acquiring data from certain hardware devices.

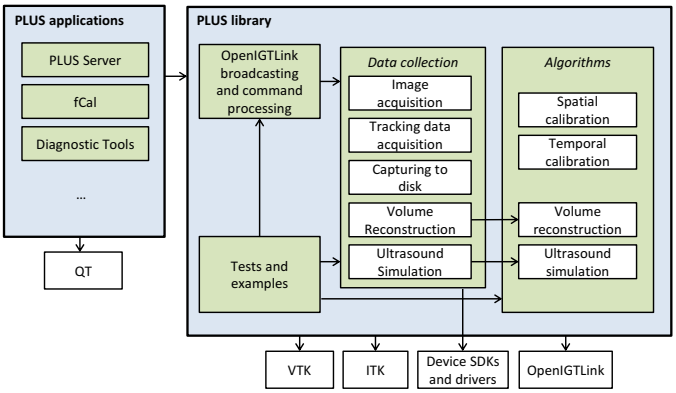

Figure 6. Software components of the PLUS toolkit. Arrows point to components that another component depends on. All modules depend on the PLUS common component, however the associated arrows are omitted for clarity.

Applications that use PLUS can be implemented similarly to existing PLUS examples, simply using the PLUS library and the Qt toolkit. However, software applications for image-guided interventions can be much more efficiently implemented by using the 3D Slicer application platform (Fig. 7). Using PLUS and 3D Slicer with a few readily available extension modules it is possible to set up image guidance application prototypes within hours, without the need for any custom software development. The OpenIGTLink connection allows a clean separation of the visualization and processing application from the data acquisition process, and the data can even be acquired on a different computer. If flexibility is not required and network throughput is a concern, then 3D Slicer modules can use the PLUS library directly to acquire all data within the 3D Slicer process.

As most of the hardware devices are supported only on Microsoft Windows and also the majority of computers in hospitals use this operating system, most of the efforts are focused on providing support for this platform. At the same time, portable solutions were chosen over operating system specific implementations wherever it was possible. Currently, all hardware-independent modules work on 32-bit and 64-bit Windows, Linux, and Mac OS.

2) File formats: While there are many file formats for 2D and 3D medical images, we could not find any widely

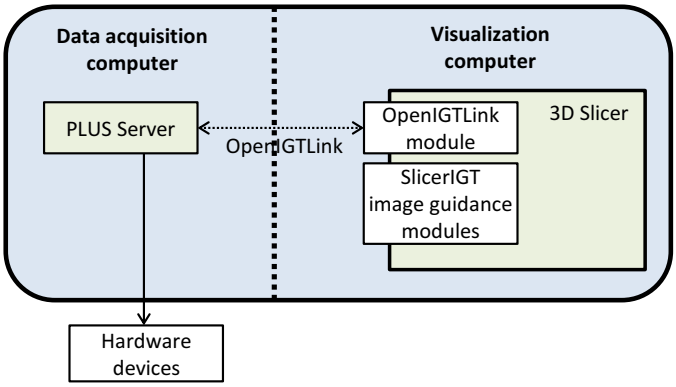

Figure 7. Building 3D Slicer [35] plug-in modules without being directly linked to PLUS, using OpenIGTLink connection.

adopted file format for the storage of tracked US data. Therefore, we chose to extend the commonly used MetaIO image (http://www.itk.org/Wiki/ITK/MetaIO) format with custom fields for storing image type, orientation, and for each frame tracking data, status, and timestamp.

There are many configuration parameters in PLUS for the specification of the data acquisition hardware devices and various processing algorithms. PLUS stores all these configuration settings in a single *Device Set* configuration file in XML format.

3) Development process and quality assurance: PLUS developers use state-of-the-art tools and best practices for maintaining and extending the toolkit. All data is stored and shared using a third-party cloud-based integrated software development collaboration service, therefore outsourcing the setup, maintenance, and development of the collaboration infrastructure. All source code, test data, CAD models, documentation, and other critical information are stored under version control. Bugfixes and feature requests are tracked using tickets. Code commits are monitored by core developers and commented by using an online code review tool. Developers and users of the toolkit can discuss questions using the web interface of the collaboration suite.

PLUS uses the CMake build system to automatically download and build the required software libraries and then build the toolkit itself. CTest is used for automatic testing of all the algorithms and major components of the toolkit. Several computers are configured to run tests after each modification and also every night, on different operating systems and build configurations. Test results are published on a CDash server. Automatic testing of the application graphical user interface is performed using Sikuli [36].

III. Results

The functionalities that are described in the Requirements section were made available in PLUS version 2.0. Data can be acquired simultaneously from any number of imaging and tracking devices. The acquired data streams are organized into channels and can be synchronized, switched, processed, and transferred to external applications. The following tracking devices are currently supported in PLUS: Ascension trakSTAR 3DG (same as the SonixGPS system built into Ultrasonix scanners) and 3DGM, NDI electromagnetic and optical trackers (Certus, Aurora, Polaris, Optotrak), BrainLab trackers (through OpenIGTLink), Claron MicronTracker, Phidgets Spatial and CHRobotics MARG sensors (only for orientation), and 3DConnexion SpaceNavigator 3D mouse. The supported image acquisition devices are: Ultrasonix US scanners (SonixRP, SonixMDP, SonixTouch, SonixTablet) through Ulterius interface (B-mode and RF-mode acquisition), BK Medical US scanners (RF-mode acquisition through CameraLink interface, with optional B-mode conversion; B-mode acquisition through OEM interface), Interson USB ultrasound scanners, Epiphan framegrabbers, Imaging Controls framegrabbers, and any Windows Media Foundation or Video for Windows compatible imaging devices. In addition to the listed hardware devices, tracking and image information can be acquired from any OpenIGTLink compatible device and can be also replayed from pre-acquired data files. Acquired imaging and tracking data can be stored in standard MetaIO image file format with custom fields.

Acquisition and free-hand spatial and temporal calibration of US image and pose tracking data is implemented in the fCal (free-hand calibration) application with a convenient graphical user interface. The user can choose between various data acquisition devices without making software changes, just by editing the Device Set configuration file. The accuracy and precision of the automatic calibration algorithm is comparable to the state-of-the-art automatic and manual methods: 0.5±0.12mm calibration reproducibility (CR) and 1.0±0.12mm point reconstruction accuracy (PRA) in the image center; and 0.69±0.31mm calibration reproducibility (CR) and 1.18±0.38mm point reconstruction accuracy (PRA) average in the image, using a probe with 3cm imaging depth at 10 MHz and an optical tracker. The details of the algorithm and quantitative results are described in detail in [30]. Quantitative evaluation of the automatic temporal calibration was performed on images received from an Ultrasonix US scanner and pose information provided by an Ascension electromagnetic tracker. The temporal calibration provided very well reproducible results: 59.8±3.0ms delay. The standard deviation value is very low considering that the accuracy of timers on a PC with a generic operating system is approximately 1ms.

Continuous real-time acquisition, broadcasting through OpenIGTLink protocol, and on-request recording and volume reconstruction of tracked image data is provided by the *PlusServer* application (Fig. 8). Recording on current desktop PC hardware could keep up with the speed of ultrasound imaging. For example, sustained recording of 1280x1024x8bit frames at 30fps was possible on a two-year-old high-end PC. Latency of the data acquisition, processing, transfer, and visualization was evaluated for an Ultrasonix SonixTouch US scanner (820x616x8bits, 13fps) and a camera device connected through Microsoft Media Foundation interface (640x480x8bit, 30fps). Another video camera was set up in a way to see the transducer, the screen of the US scanner, and the screen of a PLUS test application (that directly rendered the received images) and 3D Slicer (that rendered the images that it received from PLUS through OpenIGTLink). Then the transducer was

moved and the time elapsed between seeing the transducer moving in the physical workspace, in the US scanner screen, on a PLUS application screen, and in the 3D Slicer screen was measured. The frame rate of the camera was 30fps, therefore the latency was measured with a 33ms resolution, therefore the measurements were repeated 10 times to get more precise results. The measurement results are summarized in Table I. Reported latency values of other US guidance systems were included in the table for comparison. The OpenIGTLink transfer and visualization in 3D Slicer increases the latency by approximately 17ms. [34] reported that the OpenIGTLink transfer was 8ms for 256KB image frames. Therefore, 3D Slicer's message processing, image pre-processing, and rendering are probably responsible for approximately 8-10ms delay. Compared to a simple video input, the latency of US image display was approximately 100ms longer. As the image display on the US scanner's own screen had a latency of 134ms, this additional 100ms delay is probably can be attributed to the additional time needed for US image construction.

Table I
COMPARISON OF THE LATENCY OF THE DISPLAY OF AN US SCANNER AND VARIOUS GUIDANCE SYSTEMS. LATENCY IS MEASURED AS THE TIME DIFFERENCE BETWEEN MOVING A PHYSICAL OBJECT AND SEEING THE DISPLACEMENT ON SCREEN.

Image type	Appears on display	Latency (ms)
US (Ultrasonix)	US scanner	134
Video	PLUS test application	148
Video	3D Slicer (PLUS/OpenIGTLink)	165
US (Ultrasonix)	3D Slicer (PLUS/OpenIGTLink)	228
Video	CNMC navigation system [13]	150
US (BK)	CNMC navigation system [13]	230
US	MITK US [24]	250 or less

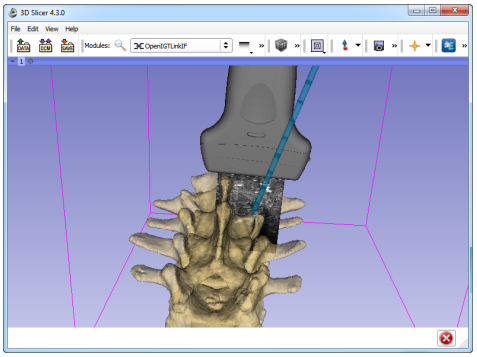

Figure 8. Screenshot of the 3D Slicer application visualizing the live US image and a tracked needle based on the information that it receives from the PlusServer through OpenIGTLink

The command-line *VolumeReconstructor* application is implemented for reconstructing volumes from tracked US frames. Typically volume reconstruction can be performed in real-time, as the US image frames are acquired, therefore at the end of the image acquisition, the reconstructed volume is already available. Computaton times obtained for 495x488x8-bit US image frames, output volume size of 500x520x360 (spacing 0.1x0.1x0.1mm) running on 4 threads on an Intel Core i5-2520M laptop are shown in Table II. The optional hole filling post-processing step on the reconstructed volumes typically takes a few seconds to complete.

Table II
PERFORMANCE OF THE VOLUME RECONSTRUCTOR

Interpolation	Compounding	Average slice pasting time (ms)
nearest neighbor	off	3
nearest neighbor	on	4.5
linear	off	12
linear	on	24

Simulation of US images was used for facet joint and centerline needle insertion training ([33]). Generation of 476x689 pixel images from 3 objects (soft tissue, needle and spine: altogether 184000 triangles) simulating a curvilinear transducer 128 with scanlines was possible at a rate of 19fps on a current desktop PC using a single thread. A sample visualization of the simulated scene and US image is shown in Fig. 9.

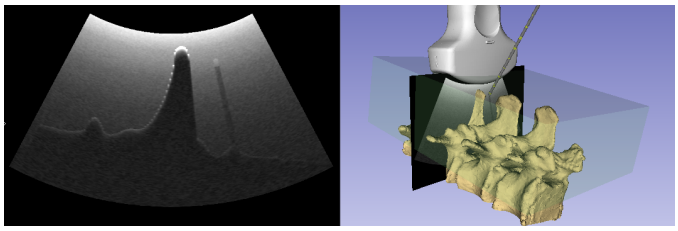

Figure 9. Screenshot of the 3D Slicer application visualizing the simulated US image and the corresponding 3D model

The targeted non-functional requirements are also fulfilled. PLUS is open-source, freely available to researchers and system developers. PLUS is distributed with a BSD-type license, which allows free, unrestricted use, although PLUS is not certified for any particular clinical application. The source code, data, documentation are freely available, without requiring registration, at the project webpage (*http://www.plustoolkit.org*). Only open standards were used for implementing the toolkit, such as XML and MetaIO for file storage and OpenIGTLink for live streaming. PLUS is easily extensible, owing to the flexibility of the data representation used and its modular design. Maintainance of the toolkit is supported by the collaboration infrastructure, the automatic testing suite, and the diagnostic tools.

Compared to custom software solutions, which support only one specific hardware configuration and processing workflow, PLUS provides about the same performance (system latency, spatial and temporal calibration accuracy, etc.), but much more flexibility and extensibility. Compared to other generic toolkits, such as IGSTK, and MITK, there are two important differences. First, PLUS does not attempt to provide a complete application framework, it only aims for performing low-level data collection and processing, therefore it does not replace these toolkits but can augment them. For example, PLUS can already receive tracker outputs from IGSTK through OpenIGTLink. MITK could receive calibrated, synchronized data streams from PLUS for further processing and visualization. The other difference is that PLUS can be used for prototyping of complete image guidance systems without *any* software development work. The hardware configuration

and processing workflow is fully specified in an XML file and no programming is needed for graphical user interface either, if PLUS is connected to software application that already contains all the potentially needed elements. Such flexible application already exist: the open-source 3D Slicer application with its SlicerIGT extension already provide all the commonly used registration, calibration, data import/export, analysis, and visualization features. The MeVisLab framework may also be used for programming-free system prototyping, as it can receive data from PLUS using OpenIGTLink ([37]). PLUS can also be used to augment CISST and MUSiiC libraries as they can send and receive each other's data streams through OpenIGTLink. We also provide an OpenIGTLink communication library for streaming data to and from pose tracking data processing algorithms implemented in Matlab® environment.

The first public version of PLUS was released in November 2011 and since then it has been used by a number of research groups and commercial companies for implementing US-guided intervention systems for translational research and product feasibility studies. In-house projects include the PerkTutor system for training and evaluation for image-guided needle insertion and navigation system for US-guided predicle screw placement, facet joint injection, and spinal curvature measurement ([38], [39], [40], [41]). PLUS is used at several research centers around the world. In the Brigham and Women's Hospital (Boston, MA, USA) PLUS is used for US-based brain shift visualization during surgery (Fig. 10) and for MRI/US fusion to help diagnosis and treatment of prostate cancer ([10]), at University of British Columbia (Vancouver, BC, Canada) for research in US-guided spine interventions [9], [14], at the Children's National Medical Center (Washington, DC, USA) for calibration of laparoscopic ultrasound probes ([13]), at the Duesseldorf University Hospital (Duesseldorf, Germany) for navigation in neck surgery ([42]) and several other groups are using or currently evaluating the toolkit (such as [12], [43], [15]).

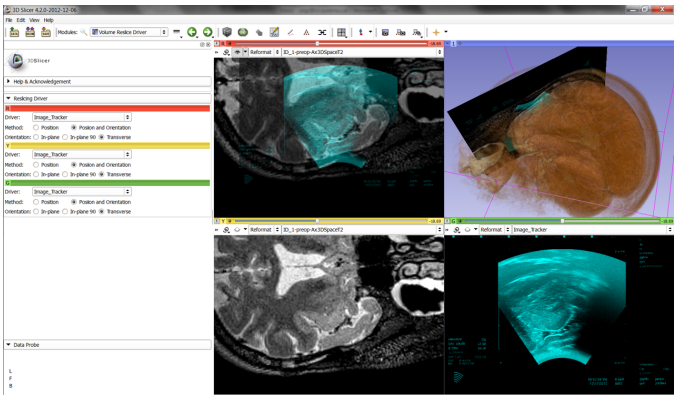

Figure 10. A screenshot of brain shift visualization in 3D Slicer using MRI and US volume fusion (Brigham and Women's Hospital: PI Wells and Aylward, Neurosurgeon Alexandra Golby, NIH Grant R01CA138419 on Image registration for ultrasound-based neurosurgical navigation). PLUS was used for acquiring and reconstructing an US volume and sending the live US image for visualization in 3D Slicer.

Since all the identified requirements have been fulfilled, we expect PLUS to become a widely used platform for US-guided intervention research and prototyping. The permissive BSD-type license allows customization or optimization of the toolkit to meet further needs. End-user applications are intended to be implemented using appropriate application frameworks, such as 3D Slicer [35] or any other OpenIGTLink-compatible software. The PLUS library and applications are under continuous improvement, which includes testing, bugfixing, and development of new features.

ACKNOWLEGMENT

This work was supported through the Applied Cancer Research Unit program of Cancer Care Ontario with funds provided by the Ontario Ministry of Health and Long-Term Care. Gabor Fichtinger was funded as a Cancer Ontario Research Chair. Tamas Ungi was supported as a Queen's University–Ontario Ministry of Research and Innovation Post-doctoral Fellow.

The PLUS toolkit reused parts of the SynchroGrab library for pose data interpolation, volume reconstruction, and interfacing with Ultrasonix, NDI, and Claron devices. The development of the SynchroGrab library was funded in part by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Canadian Institutes of Health Research (CIHR), and Canada Foundation for Innovation (CFI), principal investigators Purang Abolmaesumi and Parvin Mousavi.

The authors wish to thank the students and staff engineers at Queen's University, University of British Columbia, and Robarts Research Institute for their contribution to the toolkit.

REFERENCES

[1] A. T. Gray, "Ultrasound-guided regional anesthesia: current state of the art." *Anesthesiology*, vol. 104, no. 2, pp. 368–73, discussion 5A, Feb 2006.

[2] A. Kumar and A. Chuan, "Ultrasound guided vascular access: efficacy and safety." *Best Pract Res Clin Anaesthesiol*, vol. 23, no. 3, pp. 299–311, Sep 2009.

[3] C. J. Harvey, J. Pilcher, J. Richenberg, U. Patel, and F. Frauscher, "Applications of transrectal ultrasound in prostate cancer." *Br J Radiol*, vol. 85 Spec No 1, pp. S3–17, Nov 2012.

[4] A. Potthoff, M. J. Gebel, and K. Rifai, "[diagnostic and interventional abdominal ultrasonography]." *Internist (Berl)*, vol. 53, no. 3, pp. 261–270, Mar 2012.

[5] W. Wein, S. Brunke, A. Khamene, M. R. Callstrom, and N. Navab, "Automatic ct-ultrasound registration for diagnostic imaging and image-guided intervention." *Med Image Anal*, vol. 12, no. 5, pp. 577–585, Oct 2008.

[6] P. A. Pinto, P. H. Chung, A. R. Rastinehad, A. A. Baccala, Jr, J. Kruecker, C. J. Benjamin, S. Xu, P. Yan, S. Kadoury, C. Chua, J. K. Locklin, B. Turkbey, J. H. Shih, S. P. Gates, C. Buckner, G. Bratslavsky, W. M. Linehan, N. D. Glossop, P. L. Choyke, and B. J. Wood, "Magnetic resonance imaging/ultrasound fusion guided prostate biopsy improves cancer detection following transrectal ultrasound biopsy and correlates with multiparametric magnetic resonance imaging." *J Urol*, vol. 186, no. 4, pp. 1281–1285, Oct 2011.

[7] J. Y. Lee, B. I. Choi, Y. E. Chung, M. W. Kim, S. H. Kim, and J. K. Han, "Clinical value of ct/mr-us fusion imaging for radiofrequency ablation of hepatic nodules." *Eur J Radiol*, vol. 81, no. 9, pp. 2281–2289, Sep 2012. [Online]. Available: http://dx.doi.org/10.1016/j.ejrad.2011.08.013

[8] R. W. Prager, U. Z. Ijaz, A. H. Gee, and G. M. Treece, "Three-dimensional ultrasound imaging." vol. 224, no. 2, 2010, pp. 193–223.

[9] H. Al-Deen Ashab, V. A. Lessoway, S. Khallaghi, A. Cheng, R. Rohling, and P. Abolmaesumi, "An augmented reality system for epidural anesthesia (area): Prepuncture identification of vertebrae." *IEEE Trans Biomed Eng*, vol. 60, no. 9, pp. 2636–2644, Sep 2013.

[10] A. Fedorov, S.-E. Song, T. Kapur, L. Wolfsberger, R. Owen, I. Elliott, W. M. Wells, and C. Tempany, "Adjustable sleeve template assembly device for joint mr/us-guided diagnosis and treatment of prostate cancer: Initial design and feasibility evaluation," in *5th Image Guided Therapy Workshop, National Center for Image-Guided Therapy*, 2012, p. 11. [Online]. Available: http://www.ncigt.org/publications/item/view/2235

[11] A. Fedorov, A. Lasso, M. Moradi, S.-E. Song, E. Neubauer, R. Owen, T. Kapur, W. M. Wells, P. Nguyen, G. Fichtinger, and C. M. Tempany, "Towards open source infrastructure for joint mri/trus guided prostate interventions," in *6th Image Guided Therapy Workshop, National Center for Image-Guided Therapy*, Crystal City, VA, USA, 03/2013 2013, p. 67. [Online]. Available: http://www.spl.harvard.edu/publications/item/view/2334

[12] G. Ghoshal, T. Heffter, E. Williams, C. Bromfield, V. Salgaonkar, L. Rund, J. M. Ehrhardt, C. J. Diederich, and E. C. Burdette, "In situ treatment of liver using catheter based therapeutic ultrasound with combined imaging and gps tracking," in *Proc. SPIE Medical Imaging*, 2013, pp. 85 840T–85 840T–10. [Online]. Available: http://dx.doi.org/10.1117/12.2008258

[13] X. Kang, J. Oh, E. Wilson, Z. Yaniv, T. D. Kane, C. A. Peters, and R. Shekhar, "Towards a clinical stereoscopic augmented reality system for laparoscopic surgery," in *2nd MICCAI Workshop on Clinical Image-based Procedures: Translational Research in Medical Imaging*, ser. Lecture Notes in Computer Science, vol. 8361. Springer, 2013, pp. 108–116.

[14] A. Rasoulian, R. Rohling, and P. Abolmaesumi, "Lumbar spine segmentation using a statistical multi-vertebrae anatomical shape+pose model." *IEEE Trans Med Imaging*, vol. 32, no. 10, pp. 1890–1900, Oct 2013.

[15] C. Schneider, A. Baghani, R. Rohling, and S. Salcudean, "Remote ultrasound palpation for robotic interventions using absolute elastography." *Med Image Comput Comput Assist Interv*, vol. 15, no. Pt 1, pp. 42–49, 2012.

[16] R. J. Housden, A. H. Gee, G. M. Treece, and R. W. Prager, "Sensorless reconstruction of unconstrained freehand 3d ultrasound data." *Ultrasound Med Biol*, vol. 33, no. 3, pp. 408–419, Mar 2007.

[17] J. Boisvert, D. Gobbi, S. Vikal, R. Rohling, G. Fichtinger, and P. Abolmaesumi, "An open-source solution for interactive acquisition, processing and transfer of interventional ultrasound images," in *MICCAI 2008, International Workshop on Systems and Architectures for Computer Assisted Interventions*, 2008, pp. 1–8. [Online]. Available: http://hdl.handle.net/10380/1459

[18] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit, Third Edition*. Kitware Inc., 2003.

[19] D. Pace, D. Gobbi, C. Wedlake, J. Gumprecht, J. Boisvert, J. Tokuda, N. Hata, and T. Peters, "An open-source real-time ultrasound reconstruction system for four-dimensional imaging of moving organs," in *MICCAI 2009, International Workshop on Systems and Architectures for Computer Assisted Interventions*, 08 2009, pp. 1–8.

[20] K. Gary, L. Ibanez, S. Aylward, D. Gobbi, M. B. Blake, and K. Cleary, "Igstk: an open source software toolkit for image-guided surgery," *Computer*, vol. 39, no. 4, pp. 46–53, 2006.

[21] Z. Yaniv, P. Foroughi, H.-J. Kang, and E. Boctor, "Ultrasound calibration framework for the image-guided surgery toolkit," in *Proc. SPIE Medical Imaging*, vol. 7964, 2011, pp. 79 641N–79 641N–11.

[22] G. M. Treece, A. H. Gee, R. W. Prager, C. J. C. Cash, and L. H. Berman, "High-definition freehand 3-d ultrasound." *Ultrasound Med Biol*, vol. 29, no. 4, pp. 529–546, Apr 2003.

[23] P. J. Stolka, H.-J. Kang, and E. Boctor, "The musiic toolkit: Modular real-time toolkit for advanced ultrasound research," in *MICCAI 2010, International Workshop on Systems and Architectures for Computer Assisted Interventions*, 2010, pp. 1–11. [Online]. Available: http://hdl.handle.net/10380/3172

[24] K. März, A. M. Franz, A. Seitel, A. Winterstein, R. Bendl, S. Zelzer, M. Nolden, H. P. Meinzer, and L. Maier-Hein, "Mitk-us: real-time ultrasound support within mitk." *Int J Comput Assist Radiol Surg*, vol. 8, no. 6, pp. 1063–1072, Dec 2013.

[25] A. Kapoor, A. Deguet, and P. Kazanzides, "Software components and frameworks for medical robot control," in *Proc. IEEE Int. Conf. Robotics and Automation ICRA 2006*, 2006, pp. 3813–3818.

[26] J. von Spiczak, E. Samset, S. DiMaio, G. Reitmayr, D. Schmalstieg, C. Burghart, and R. Kikinis, "Multimodal event streams for virtual reality," in *Proc. SPIE Medical Imaging*, vol. 6504, 2007, pp. 65 040M–65 040M–8. [Online]. Available: http://dx.doi.org/10.1117/12.706079

[27] R. M. T. T. C. Hudson, A. Seeger, H. Weber, J. Juliano, and A. T. Helser, "Vrpn: A device-independent, network-transparent vr peripheral system," in *Proceedings of the ACM Symposium on Virtual Reality Software & Technology 2001, VRST*, 2001, pp. 15–17.

[28] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*, 2nd ed., Kitware, Inc. ISBN 1-930934-15-7, http://www.itk.org/ItkSoftwareGuide.pdf, 2005.

[29] L. Mercier, T. Langø, F. Lindseth, and L. D. Collins, "A review of calibration techniques for freehand 3-d ultrasound systems." *Ultrasound Med Biol*, vol. 31, no. 2, pp. 143–165, Feb 2005.

[30] G. Carbajal, A. Lasso, A. Gómez, and G. Fichtinger, "Improving n-wire phantom-based freehand ultrasound calibration." *Int J Comput Assist Radiol Surg*, Jul 2013.

[31] T. K. Chen, A. D. Thurston, R. E. Ellis, and P. Abolmaesumi, "A real-time freehand ultrasound calibration system with automatic accuracy feedback and control." *Ultrasound Med Biol*, vol. 35, no. 1, pp. 79–93, Jan 2009.

[32] D. G. Gobbi and T. M. Peters, "Interactive intra-operative 3d ultrasound reconstruction and visualization," in *MICCAI (2)*, 2002, pp. 156–163.

[33] L. Bartha, A. Lasso, C. Pinter, T. Ungi, Z. Keri, and G. Fichtinger, "Open-source surface mesh-based ultrasound-guided spinal intervention simulator." *Int J Comput Assist Radiol Surg*, vol. 8, no. 6, pp. 1043–1051, Nov 2013.

[34] J. Tokuda, G. S. Fischer, X. Papademetris, Z. Yaniv, L. Ibanez, P. Cheng, H. Liu, J. Blevins, J. Arata, A. J. Golby, T. Kapur, S. Pieper, E. C. Burdette, G. Fichtinger, C. M. Tempany, and N. Hata, "Openigtlink: an open network protocol for image-guided therapy environment." *Int J Med Robot*, vol. 5, no. 4, pp. 423–434, Dec 2009.

[35] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J.-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S. Aylward, J. Miller, S. Pieper, and R. Kikinis, "3d slicer as an image computing platform for the quantitative imaging network," *Magnetic Resonance Imaging*, vol. 30, no. 9, pp. 1323–1341, 11 2012.

[36] T. Yeh, T.-H. Chang, and R. C. Miller, "Sikuli: using gui screenshots for search and automation," in *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, ser. UIST '09. New York, NY, USA: ACM, 2009, pp. 183–192.

[37] J. Egger, J. Tokuda, L. Chauvin, B. Freisleben, C. Nimsky, T. Kapur, and W. Wells, "Integration of the openigtlink network protocol for image-guided therapy with the medical platform mevislab." *Int J Med Robot*, vol. 8, no. 3, pp. 282–290, Sep 2012.

[38] T. Ungi, D. Sargent, E. Moult, A. Lasso, C. Pinter, R. C. McGraw, and G. Fichtinger, "Perk tutor: An open-source training platform for ultrasound-guided needle insertions," *IEEE Trans Biomed Eng*, vol. 59, pp. 3475–3481, 12/2012 2012.

[39] T. Ungi, E. Moult, J. H. Schwab, and G. Fichtinger, "Tracked ultrasound snapshots in percutaneous pedicle screw placement navigation: a feasibility study." *Clin Orthop Relat Res*, vol. 471, no. 12, pp. 4047–4055, Dec 2013.

[40] E. Moult, T. Ungi, M. Welch, J. Lu, R. C. McGraw, and G. Fichtinger, "Ultrasound-guided facet joint injection training using perk tutor." *Int J Comput Assist Radiol Surg*, vol. 8, no. 5, pp. 831–836, Sep 2013.

[41] T. Ungi, F. King, M. Kempston, Z. Keri, A. Lasso, P. Mousavi, J. Rudan, D. P. Borschneck, and G. Fichtinger, "Spinal curvature measurement by tracked ultrasound snapshots." *Ultrasound Med Biol*, vol. 40, no. 2, pp. 447–454, Feb 2014.

[42] N. Jansen, T. Brennecke, J. Hirschfeld, L. Colter, J. Raczkowsky, H. Woern, and J. Schipper, "Rotatable flexible neck-model for the evaluation of minimally invasive operation procedures with the help of an ultrasound-based navigation system." vol. 2013, 2013, pp. 1140–1143.

[43] V. Shivaprabhu, A. Enquobahrie, Z. Mullen, and S. Aylward, "Accelerating ultrasound image analysis research through publically available database," in *Proc. SPIE Medical Imaging*, vol. 8675, 2013, p. 867517.