ORIGINAL ARTICLE

MITK-US: real-time ultrasound support within MITK

K. März · A. M. Franz · A. Seitel · A. Winterstein · R. Bendl · S. Zelzer · M. Nolden · H.-P. Meinzer · L. Maier-Hein

Abstract

Purpose Intra-procedural acquisition of the patient anatomy is a key technique in the context of computer-assisted interventions (CAI). Ultrasound (US) offers major advantages as an interventional imaging modality because it is real time and low cost and does not expose the patient or physician to harmful radiation. To advance US-related research, the purpose of this paper was to develop and evaluate an open-source framework for US-based CAI applications.
Materials and methods We developed the open-source software module MITK-US for acquiring and processing US data as part of the well-known medical imaging interaction toolkit (MITK). To demonstrate its utility, we applied the module to implement a new concept for US-guided needle insertion. Performance of the US module was assessed by determining frame rate and latency for both a simple sample application and a more complex needle guidance system.
Results MITK-US has successfully been used to implement both sample applications. Modern laptops achieve frame rates above 24 frames per second. Latency is measured to be approximately 250 ms or less.
Conclusion MITK-US can be considered a viable rapid prototyping environment for US-based CAI applications.

K. März (✉) · A. M. Franz · A. Seitel · A. Winterstein · R. Bendl · S. Zelzer · M. Nolden · H.-P. Meinzer · L. Maier-Hein
German Cancer Research Center (DKFZ), Im Neuenheimer Feld 280, 69120 Heidelberg, Germany
e-mail: k.maerz@dkfz-heidelberg.de; k.maerz@dkfz.de

L. Maier-Hein
e-mail: l.maier-hein@dkfz.de

R. Bendl
Medical Informatics, Heilbronn University, Max-planck-str. 39, 74081 Heilbronn, Germany

Introduction

> We argue that, with some exceptions, anything less than the release of source programs is intolerable for results that depend on computation.
> – Ince, Hatton, Graham-Cumming (Nature 2012) [6]

In the past decade, the scientific community has gradually become aware of the importance of open-source software for science, as stated by a recent *Nature* paper [6]. The medical imaging interaction toolkit (MITK[1]) is an open-source framework that was originally designed as a platform for medical image processing [21]. Over the last decades, the field of computer-assisted interventions (CAI) has made significant progress [1]. Following this trend, several CAI applications have been developed for MITK. As a result, modules such as MITK-IGT [5] which is used for developing image-guided therapy (IGT) applications and MITK-ToF [18] for controlling range image devices such as Time-of-Flight (ToF) cameras made key functionality for navigation applications available.

While previous works in the field of CAI focused on marker-based registration of preoperative image data (e.g., [9,12]), there is a current trend toward using real-time interventional imaging in CAI [14]. Ultrasound (US) is a good candidate modality in several respects: Its high availability, real-time image stream and radiation-free application make it easy to work with, while the advanced

---

[1] http://www.mitk.org.

Fig. 1 Screenshot of the MITK-US-based application *UltrasoundSupport*, which allows us to manage and view images from US devices. The upper marked area shows available plugins. The right marked area shows the user interface of the ultrasound plugin, with B-Mode controls for an APl US devices
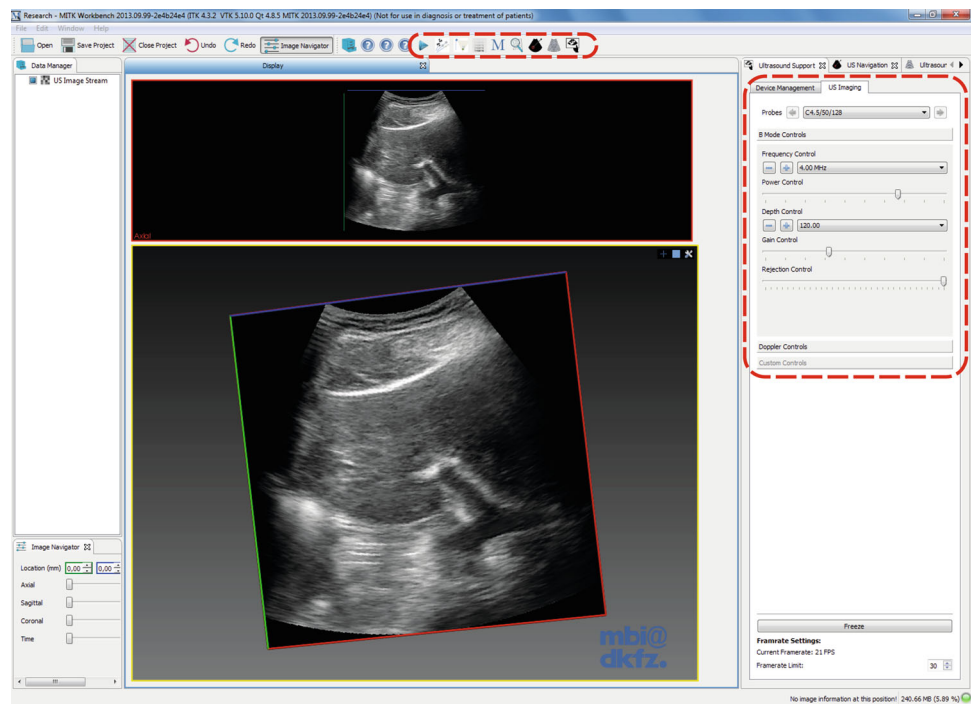
state of US technology allows for a reliable and relatively low-cost usage. The uncomplicated application of US has led to its embedding in many everyday workflows where physicians rely on the fast and problem-free application. This also makes US an ideal imaging modality for needle guidance applications. On the other hand, the image quality of US is low, making it difficult for users to discern features such as target or critical structures and tools such as needles. Different solutions have been proposed to enhance visibility through, e.g., image fusion [2], or to facilitate interventions, e.g., through needle guidance [10]. This shows that US is not only used as a pure imaging modality but is increasingly integrated into image-guided therapy applications. Facilitating the prototyping of robust US-based CAI applications is thus of major importance. Existing open-source solutions either do not support real-time processing of US data [20] or are designed as complete CAI toolkits [7] with additional functionality such as tracking support, which makes it inefficient to be integrated into MITK regarding, e.g., issues such as code duplication.

This paper presents and evaluates a new open-source module for computer-assisted US interventions (MITK-US, see section "MITK-US"). Requirements and architecture are shown in this chapter as well. Afterward, an example implementation for a navigation system is shown in "Example application: US-guided navigated needle insertion" section. We conduct an extensive performance analysis and present our assessment in "Results" section and discussed in "Discussion" section.

MITK-US

MITK-US was designed and developed by applying the MITK software processes and modularization methods [16]. We first present the requirement analysis ("Requirements" section) used to derive the architectural design, which will then be explained in "Architecture" section (Fig. 1).

Requirements

The aim for this development was to create an environment that can be used for rapid prototyping of US-based CAI applications. MITK-US should be able to be used as a basis for the development of application used in, e.g., clinical trials. To this end, the following goals were defined:

Wide hardware support: It must be possible to connect and use everyday US devices as well as modern machines with a dedicated Application Programming Interface (API).

Extensibility: The module must be extensible for future development like the use of real-time 3D US.

Flexibility: A flexible concept for real-time data processing must be present that makes it easy to quickly exchange or add components.

Performance: MITK-US must deliver high performance regarding frame rate and latency where real-time issues are present. At the same time, integration of time-consuming processing steps must not be impeded, if they are required by the developer.

Application-wide availability: The module should make configured devices available to other plugins in a service-oriented architecture approach [16]. Workflows should be reusable for other tasks once programmed. For example, one plugin can handle connection and image acquisition, while several other plugins use connected devices for their own purposes.

Portability: The software must be runnable on Windows, Mac and Linux systems.

Robustness: Development should happen under a suitable and defined development process including continuous testing to ensure code quality. This is also an important concern for safety in applications using the toolkit. Complete testing ensures correct results and prevents many unforeseeable errors.

Usability: The toolkit must be easy to use for developers and consumers alike.

Accuracy: A software framework for CAI must ensure the correct treatment of positional data and spatial relations.

Architecture

MITK-US is the new US module for MITK (see Fig. 2) and contains the necessary abstraction for diverse US devices. It uses OpenCV[2] to acquire images and to manipulate them. MITK-USUI is a supporting module that makes several Qt[3] widgets available, which allow users to interact with the module functionality and developers to quickly add basic US functionality to new applications. MITK-US is structured in a layer architecture which is shown in Fig. 3.

Hardware layer

The hardware layer abstracts the communication with the US hardware and allows for real-time US image acquisition. A US device is responsible for the communication with the hardware and provides an US image stream to the processing layer (see Fig. 3). The module is designed to support an extendable variety of US devices. These can be newer models that offer a software development kit (SDK) to communicate with the device directly, or virtual devices that abstract specific tasks or workflows. An example for such a specific workflow is the application presented in "Example application: US-guided navigated needle insertion" section. If needed, a device can be placed in a separate subproject and manage its own dependencies. A device that requires tracking capability, for example, can import functionality from the module MITK-IGT.

---

[2] http://www.opencv.org.
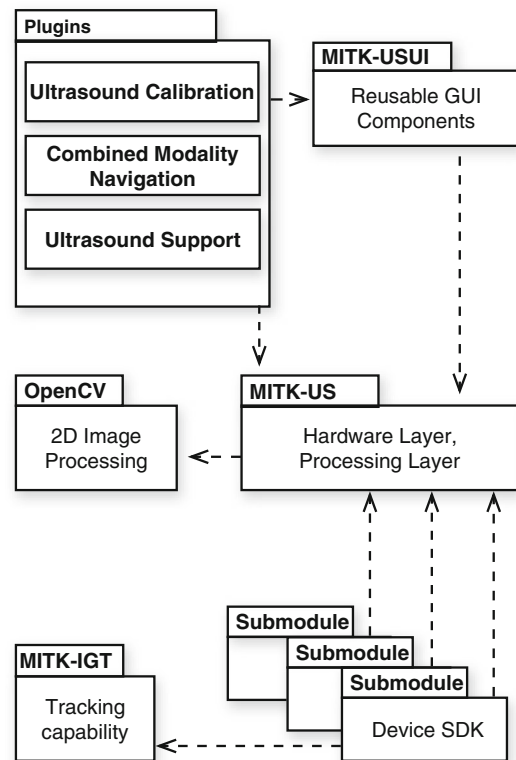
[3] http://qt.digia.com/.

Fig. 2 Module structure of MITK-US. The module contains the central code to US imaging and depends on OpenCV for image acquisition and 2D preprocessing. Device-specific submodules can implement proprietary SDKs for direct communication with the devices. They can also depend on other modules such as MITK-IGT [5] if their functionality is necessary. MITK-USUI provides reusable widgets for common interaction tasks. The plugins cover sets of use cases to achieve specific tasks

Modern US devices sometimes offer an API to allow communication between computer and device. These interfaces commonly come in the form of proprietary SDKs, which can then be implemented in separate submodules. However, such API-enabled devices are rare in a clinical environment. Most devices possess a video output which is used to connect secondary equipment such as additional screens or a video recorder. The signal of this video output can be captured via a video grabber to redigitalize it. Older US devices usually feature a S-Video or triple RCA jack output which are limited to relatively low resolutions and often carry an interlaced signal. Video grabber for these outputs is typically connected to the PC via USB and is low cost and easily obtainable. Modern US devices are usually equipped with VGA, DVI or HDMI connectors. These provide high signal quality and resolution, but also require more sophisticated hardware. Matching video grabber is available for USB and PCI connections. The bandwidth of USB 2.0 can limit the frame rate of higher-resolution signals of above 1024×768 pixel. Newer
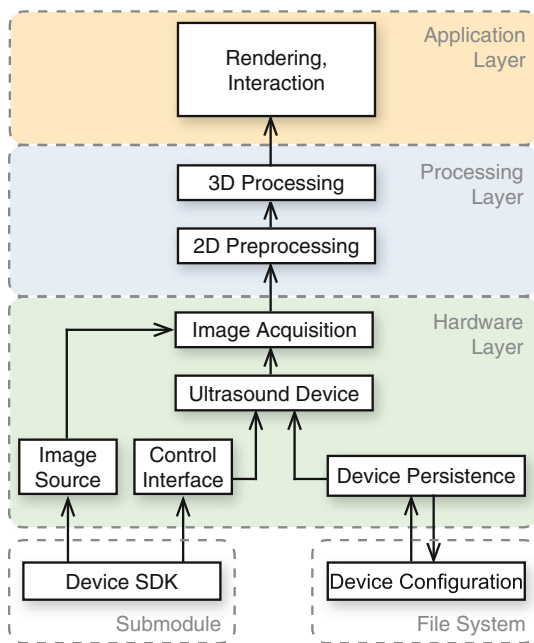
Fig. 3 Layer structure of MITK-US. The hardware layer uses the device SDKs in the submodules to implement a control interface that translates interaction commands, e.g., zoom or freeze, for the ultrasound device. An image source can also be implemented by the submodule, or images can be acquired by a video grabber. These objects are used by the common device superclass that can in turn be used by a wide set of applications. Acquired images are passed through the processing layer to be rendered for the user in the application layer. If a device has no SDK available, then no control interface is implemented and the video grabber is used as an image source

USB 3.0 compatible devices do usually not have this limitation.

The parent class of all US devices is `mitk::USDevice` which implements practically all tasks to function in the MITK environment. US devices that are connected via a video grabber can be used as a `mitk::USVideoDevice` directly. Devices with API access need to be subclassed and connected to their proprietary API by implementing the control interface (see Fig. 3). Each SDK is unique, and the interface provides a common means of communication for all applications.

To avoid repetitive configuration steps, each device is persisted once configured and available at any subsequent startup until it is removed manually. The *Device Persistence* (see Fig. 3) will save any configuration that can be persisted in a meaningful way. Parameters vary between devices and can include connection settings, the last mode used or spatial information on the US image.

Additionally, devices are implemented as so-called *Microservices*, a concept that is based on the Service Layer specifications of the *Open Services Gateway initiative* (OSGi). It is used in MITK to make service classes available across the application in a service-oriented architecture

approach. Once a device is configured and connected, other applications can use it to obtain images. This allows easy reusability of configured devices for new applications. For example, configuration of devices can be done in the plugin *Ultrasound Support* (see Fig. 2), and new applications can simply access the devices. Furthermore, if two devices are implemented that support different tracking paradigms and thus different methods of calibration, an application that requires tracked ultrasound can use either of them without requiring adaption. The benefits of this concept become clear when comparing it to previous advances. Before implementing devices as microservices, each application that wanted to use ultrasound would have had to implement its own code for connection logic, which results in a significant duplication of code. Furthermore, when improvements to the method of connection were performed, these changes had to be brought to each plugin separately. Now changes in one place are sufficient. The concept of microservices is rather complex; for further information, we refer the reader to Nolden et al. [16].

Processing layer

The processing layer provides a structure for pipelines to receive and manipulate images. The standard configuration consists of a fast 2D preprocessing part and a connected 3D pipeline (see Fig. 4). After the image is acquired, it is first fed into the 2D preprocessing part, where image manipulation can happen very fast with the help of OpenCV. The fact that image processing via OpenCV happens very efficiently but only in 2D makes this part of the pipeline optimal for preprocessing steps.

Since the video output of most US devices contains additional screen items such as patient information, the image is first cropped to the actual US image. To speed up further processing steps, it is then converted to gray scale, as capture devices always capture in color even if the input is gray scale. The 2D preprocessing section can be modified freely: Steps
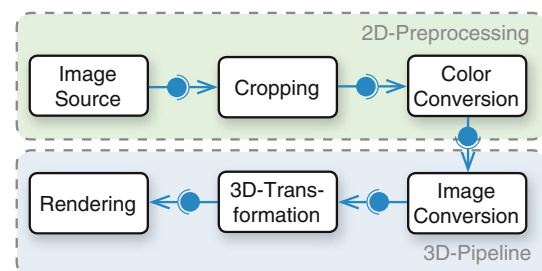

Fig. 4 Processing layer of MITK-US. This is the typical minimal pipeline configuration. In the 2D preprocessing section, the image is cropped and converted to gray scale. It is then transformed to a 3D image and fed into the 3D pipeline, transformed into 3D space and then rendered. Both 2D preprocessing and 3D pipeline can be extended or shortened

can be removed, for example, when colored Doppler images are captured, and steps can also be added, for example, when 2D segmentation is required.

After passing through the 2D preprocessing part, the images are converted into an `mitk::Image`, which is the basic data structure for images of MITK, and then enter the 3D Pipeline. The pipeline concept is derived from the underlying Insight Segmentation and Registration Toolkit and provides a means to extend data processing workflows by using custom filters that manipulate data passing through them. The image is transformed into 3D space according to the device calibration. Again, the pipeline is fully flexible. For example, the 3D transformation can be left out if there is no need to transform the image to another coordinate space. At this stage all filters of MITK can be applied, which gives the developer access to a wide range of functionality. The application also defines whether the image stream is rendered or, for example, saved to disk via a file writer.

Application layer

The final layer is realized by plugins and the MITK-USUI module [16]. To simplify application prototyping with MITK-US, basic graphical user interface (GUI) elements have been designed to be reusable in many US-based applications reducing source code duplication. For example, the plugin *UltrasoundSupport*, which can be seen in Fig. 1, provides a means to configure and manage available devices and to stream images from them. Plugins can also connect to the pipeline endpoints from US device to continue processing data. At the end, the image data are usually fed to a sink like rendering or a file writer to record images.

Performance analysis

To measure the performance of MITK-US in a realistic scenario, we determined the frame rates of both the 2D and the 3D processing part (see Fig. 4). For the 2D part, the time from image acquisition to gray scale conversion was measured for frame rate computation. For the 3D part, the time from the creation of an mitk:USImage to the rendering was determined. Note that the overall frame rate after the 3D pipeline may be limited by one of the 2D processing in cases where 2D processing takes longer than 3D processing. For this evaluation, we thus allowed the 3D pipeline to process images multiple times to be able to achieve a separate 3D frame rate. This was done to assess the bottleneck for each step and to give an impression of where optimizations are to be prioritized.

To assess these parameters, two systems were used. One was a personal computer from 2011 (Intel i7-870 4 ×

2.93 GHz, 8 GB Ram, NVIDIA GeForce GTX 470), employing a USB 2.0 port. The second was a current high-end laptop (Intel i7-2860QM, 4 × 2.50 GHz, 16 GB RAM, NVIDIA Quadro 2000M) using an USB 3.0 port to connect the grabber. The first system represents a mainstream system, while the second is ranked at the high-end scale of today's hardware. By using these two systems, it is possible to estimate from the results how much to invest in hardware when developing an application with MITK-US. To acquire images, we used a ZONARE Z.one US device (ZONARE Medical Systems Inc., Mountain View, California), which was connected via a HDMI/DVI cable to a Epiphan DVI2USB 3.0 grabber (Epiphan Systems Inc., Ottawa, Canada). The Z.one delivers a resolution of 1280 × 1024 px. To grab the image stream, we used an Epiphan DVI2USB 3.0 grabber, which costs 535€ (699$) at time of printing. The US image was cropped to a resolution of 680 × 680 px. Once the grabbing started, the time was measured for 100 frames processed by the respective pipeline and the measurement repeated 30 times without additional workload on the computer.

Example application: US-guided navigated needle insertion

Ultrasound image processing as provided by MITK-US can benefit CAI systems wherever US is used during an intervention. For example, in the case of percutaneous punctures, it is a priority to reach the target accurately without damaging critical structures like the vena portae or the gall bladder. This generally poses several problems such as the placement of the tracking hardware [4,8] or the tracking of critical structures.

In a previous paper [10], we proposed a new approach to US-based needle guidance based on a singular handheld modality that provides both imaging and tracking capability. This is achieved by combining a new, compact and mobile electromagnetic field generator (FG) by Northern Digital Inc. (NDI, Waterloo, Canada) with a US probe (see Fig. 5). The FG generates a tracking volume around the image plane of the US probe and through the tight integration between those two; the spatial relation between image plane and tracking volume is constant.

When tracked needles are introduced into this field, their pose is measured by the system and their path is projected along the needle axis (see Fig. 6). The intersection of projected needle path and image plane is then marked on the image. For a detailed description of the concept, the interested reader may refer to [10].

In the following, we show how MITK-US was used to implement this navigation system for percutaneous needle
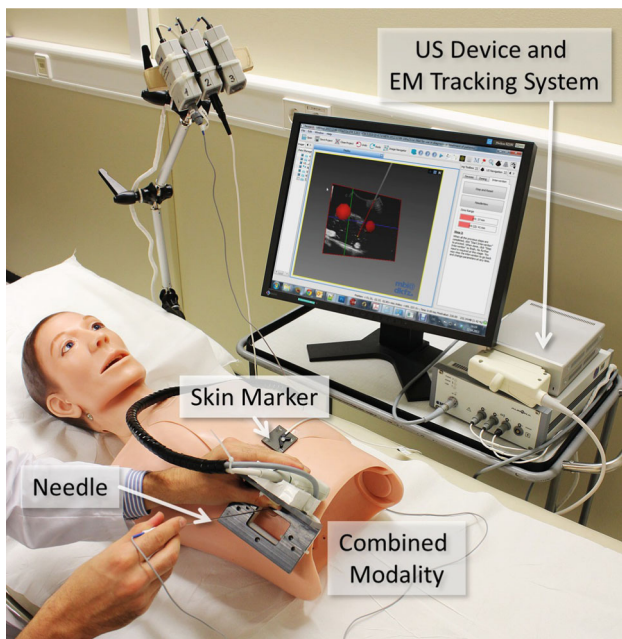
Fig. 5 A demonstration setup of the navigation system for which MITK-US provided a software implementation
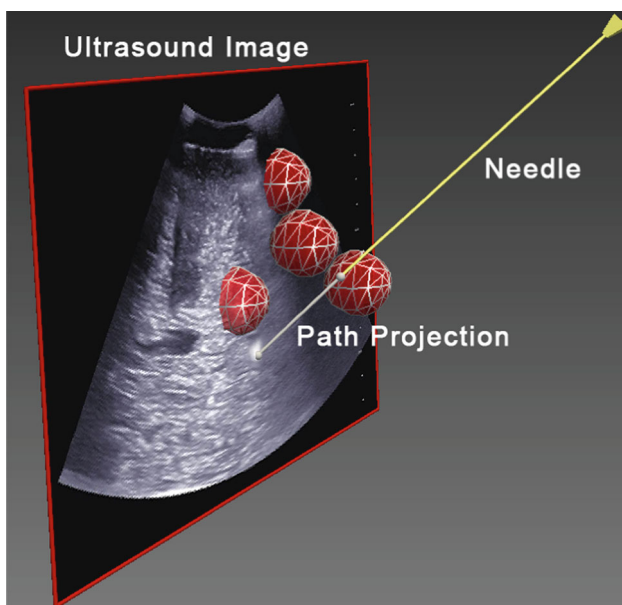

Fig. 6 The needle guidance system with the combined modality. Needle and US image plane are rendered along with a path projection toward the image plane. The *red spheres* represent areas of risk that are managed by the plugin

insertions. This implementation is then used to quantify the performance of MITK-US in an application scenario.

System design

MITK-US was used for the implementation of this needle guidance solution based on the new modality described in

"Example application: US-guided navigated needle insertion". Since tracking capability is required, MITK-IGT was utilized as well. We wanted to abstract both tracking and imaging functionality into a single device that manages the required pipelines. Consequently, both US support and tracking capability were combined into a virtual device (`mitk::CombinedModality`) that is located in a sub-module of MITK-US. It abstracts common functionality for the modality and provides synchronized and calibrated data for further tasks like path projection.

Processing of US and tracking data is depicted in Fig. 7. The US images are preprocessed as described in "Architecture" section. The tracking data are smoothed by a *smoothing filter* and then synchronized with the US image stream by applying a temporal calibration (see "Performance analysis").

Once synchronized, the image is transformed into the 3D space of the tracking system by applying a spatial calibration in the *3D Transform Filter*. Several ways to calibrate US images and tracking systems have been presented [11]. Any approach that allows us to compute the pose of the image plane inside the tracking volume can be used for this task. We decided to use a point-based approach based on a modified version of the protocol from Muratore and Galloway [13]. The modality is positioned in a water bath, and a tracked needle is brought into the image plane orthogonally in a defined pattern. Once the needle is observed, the point is marked on the image plane. After marking nine dots distributed evenly throughout the plane, a point-based registration is performed to determine the calibration parameters.

Lastly, the tracking data stream passes the *needle projection filter*, which projects the needle trajectory onto the image plane and draws the predicted intersection point onto the image, which allows the physician to predict the path that the needle will travel.

Performance analysis

The performance of the guidance system implemented with MITK-US is determined in terms of the frame rate, total latency, the tracking latency and the latency divergence as detailed in the following paragraphs.

Frame rate

The same measurements taken for MITK-US were repeated for the system to see how the additional workload affects the frame rate. This case used the pipeline system shown in Fig. 7 for evaluation.
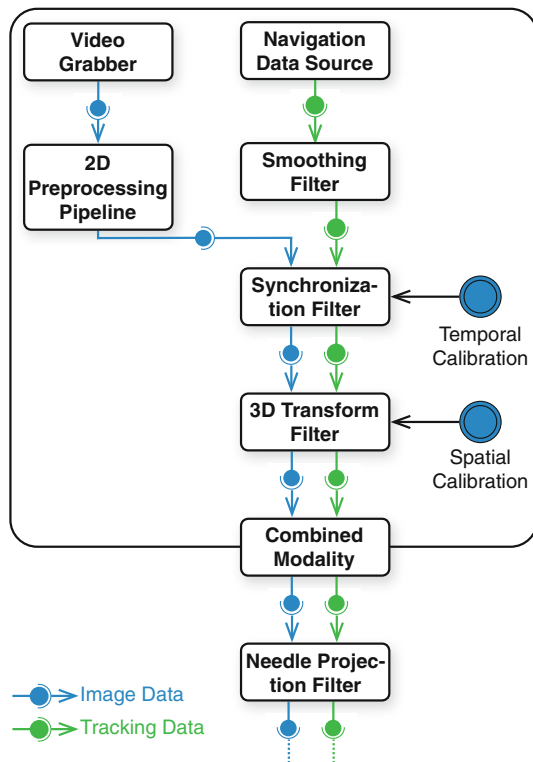
Fig. 7 The pipeline system of the combined modality. Tracking data streams are shown in *green*, and image streams are shown in *blue*. Combined Modality acts as a source for both streams for further processing

Total latency

In the context of US imaging, latency is the time between when an image is acquired by the US device and the time it is displayed to the user. Furthermore, when developing CAI applications, it is usually necessary to provide tracking data, for example, from the instruments. This means that the image stream and the tracking data stream have to be synchronized. The total latency of the system can be divided into two parts: first, the delay between the actual movement of the probe and the arrival of tracking data correlating with that movement—the *tracking latency*—and second, the delay between the arrival of tracking data and the arrival of the corresponding US image—the *latency divergence* between image and tracking data.

Tracking latency

The delay of the tracking systems is gathered or estimated from literature [3,19,22]. Since optical and electromagnetic tracking systems are two of the most used ones, we used both types of system for evaluation. The latency of the NDI Polaris system is estimated to be 70–100 ms [3,19]. It is known that optical tracking generally has a lower latency than EM tracking [22], but data on the exact latency are not available in

publications. It was relayed to us by NDI that delays of 150–180 ms could be expected; however, we used both tracking types to evaluate whether comparable total latencies occur, which would be the expected case.

Latency divergence and temporal calibration

To evaluate the latency divergence between image and tracking stream, a simple temporal calibration technique analogous to the temporal calibration tool of the PLUS toolkit, fCal [7], was used. A tracking tool was affixed to the US probe. The probe was then tracked while it was moved upward and downward in a water bath. In the image stream, the bottom of the water bath can then be observed moving upward and downward. Both tracking and image stream were recorded. The temporal divergence was calculated by identifying the high and low points of the movement in both image and tracking stream and correlating them. Using optical and EM tracking, the temporal calibration was performed five times, each time measuring five cycles with ten peaks.

Results

MITK-US was developed as described in "MITK-US". Figure 1 shows MITK-US displaying an image stream from a US device. A 2D and a 3D representation of the image can be seen. The module is open source and available as part of the MITK source distribution.[4]

The needle guidance system described in 3 has successfully been implemented with MITK-US and is described in the following. In a related study, several experiments have been performed with the navigation system [10], achieving hit rates of 92 % and an average accuracy of 3.4 ± 1.2 mm. A screenshot can be seen in Fig. 6, where an tracked needle is oriented versus an US image, with the path being projected onto the image plane.

The results of the performance analysis of MITK-US are shown in Table 1a. It can be seen that the 2D preprocessing section performed equally fast on both systems with an average of about 35 fps on both systems. However, the 3D pipeline was faster on the new system (38.8 fps) than on the older system (19.7 fps).

The results for the frame rate measurement for the pipeline system shown in Fig. 7 are shown in Table 1b. The 2D preprocessing does not seem to be impacted by the higher workload of the complex pipeline and still shows frame rates of around 35 fps. The 3D pipeline, however, is considerably slower on both systems, with a 42.3 % decrease for the *System A* and a 25.5 % decrease for the *System B*.

---

4 http://www.mitk.org/Download.

Table 1 Results from the frame rate measurement

	System A	System B
(a) Frame rates of a basic pipeline setup according to Fig. 4. No complex calculations are necessary		
2D Preprocessing	35.3 ± 0.1 fps	35.6 ± 0.2 fps
3D Pipeline	19.7 ± 1.4 fps	38.8 ± 0.9 fps
(b) Frame rates of the more complex pipeline according to Fig. 7. The image went to extensive postprocessing before it was rendered		
2D Preprocessing	35.2 ± 0.1 fps	35.4 ± 0.1 fps
3D Pipeline	11.3 ± 2.3 fps	28.9 ± 1.1 fps

Values are averages over 30 sets of 100 recorded images. *System A* is a machine from 2011, *System B* a more modern system from 2013

Table 2 Temporal divergence and absolute latency of the complex pipeline system shown in Fig. 7

Type	Tracking L. (ms)	Divergence (ms)	Total latency (ms)
Optical	70–100	113 ± 27	$\sim$200–225
EM	150–180	65 ± 5	$\sim$215–245

Tracking L. is the latency of the tracking device itself according to literature [3,22]. *Divergence* is the amount of time the image arrives after its corresponding tracking signal. *Total L.* gives an estimation of the total latency based on these numbers. $n = 10$

The results for the latency evaluation are shown in Table 2. Similar total latencies are achieved for the different tracking technologies. For optical tracking, the temporal divergence was 113 ms, leading to an estimated total latency of 200–225 ms. For electromagnetic tracking, the delay was shorter with an average of 65 ms, which lies in the expected range according to Wu et al. [22], leading to an estimated total latency of 215–245 ms. Both total latencies lie in comparable intervals.

Discussion

In this paper, we have presented the new MITK module MITK-US for US-based applications and applied it successfully for implementation of a previously proposed concept for a needle guidance system based on a combined modality for US imaging and EM tracking. Experiments confirmed an acceptable frame rate and latency of the framework, showing that development of complex applications is feasible.

As part of the Insight Segmentation and Registration Toolkit (ITK[5]), an open-source library for ultrasound image processing is available [20]; however, it does not include real-time imaging functionality. Another very promising open-source software framework is the *Public Software Library for UltraSound Imaging* (PLUS) [7], which among others has

Table 3 Comparison of the main characteristics between MITK-US and PLUS

	MITK-US	PLUS
Usage	Module for MITK	Standalone
Integration	Integrated	Via OpenIGTLink
Tracking	Via MITK-IGT	Integrated
API-Devices	Supported	Supported
OS	Windows, Mac, Linux	Windows, (Linux[a])

[a] Linux is partially supported

made available advanced calibration techniques for both temporal and spatial calibration [7]. A comparison of the main properties of MITK-US and PLUS can be seen in Table 3. First considerations to fully integrate PLUS into MITK have been rejected due to concerns regarding code duplication, especially with respect to tracking support. PLUS carries with it a number of libraries that include functionality already present in MITK, for example, a tracking framework. Duplicating this functionality in MITK should be avoided to keep the number of dependencies low.

However, several concepts such as the temporal calibration were taken from the PLUS framework, and a partial integration of PLUS into MITK-US is still under discussion.

It was our aim to create a flexible and effective environment for the development of CAI applications for US. This was achieved by implementing a module for MITK, which allows for an easy integration into an application with defined user interface design patterns [16]. By moving devices into their own submodules, the dependencies to other modules are kept flexible and developers can freely decide what functionality they want to use. Also, the pipeline system is fully flexible, as it can be configured to every use case.

However, the module structure also gives the developer access to many preexisting modules and their functionality. This is demonstrated with the quick and effective implementation of `mitk::CombinedModality`. It was possible to quickly create a device that allows tracking of needles against its US image plane by only importing functionality from MITK-IGT and setting up a pipeline with custom filters. The common device superclass already contained all the logic for the device to be used application-wide in MITK. This allows a problem-oriented approach to creating CAI solutions for US, as the developer can abstract solutions for certain problems into reusable filters and make use of existing filters.

Any US device with an video output can be connected to MITK-US by means of a video grabber. If the device has support for an API, it can be accessed by implementing a control interface that translates calls from MITK-US for the proprietary SDK. This way, a wide range of hardware is supported.

The frame rate is important in the perception of a real-time workflow. The human brain begins to perceive sequentially displayed images as motion starting from 16 fps [17]. We assume that the current cinematic standard of 24 fps is sufficient for an immersive experience, which is also assumed in literature [17]. When grabbing images from a video input, several factors limit the final frame rate. First, the video grabber can only operate up to a certain frame rate, which is usually dependent on the resolution of the incoming signal and the bandwidth of the signal chain to the computer. Second, the image is typically processed before it is displayed, which can limit the frame rate if complicated operations are necessary. Finally, the images need to be rendered, which can reduce the frame rate, especially if it is displayed in a 3D context as given here. Table 1a shows that both systems performed equally well in preprocessing. System B, featuring a faster processor and better RAM, was significantly faster in the 3D pipeline, however. Our measurements show that the frame rates are sufficient for a fluid presentation of US images on a modern system. In many cases, the video grabber will be the bottleneck. The Epiphan DVI2USB 3.0 that we used can be considered a high-end grabber and is limited to 30 fps at a resolution of 1280×1024, which was used in our experiments. Even when using complicated pipeline setup, a modern PC is able to display more than 24 fps.

Our estimated total latency generates results in approximately the same intervals, which is a good indicator that the measurements are conclusive. A conservative assumption is a total latency of 250 ms and lower. This delay is noticeable, and it must be decided in an application-specific manner whether it is acceptable. Typical applications of ultrasound do not require rapid hand–eye coordination, in which case the delay can be considered unproblematic. It seems that most of the delay is introduced by the video grabber, which can be confirmed when viewing applications that display the grabbed image to the user directly.

The microservice approach, following concepts by OSGi, is implemented in MITK-US [16]. After creation and configuration, US devices are available throughout the application, which maximizes reusability. Once the combined modality is set up, it can be used from any plugin that requires access to its US and/or tracking data. Lastly, the platform MITK guarantees portability to the major platforms Windows, Linux and Mac and robustness by a continuous testing approach [16]. MITK-US is tested against a large range of conditions to ensure code quality, especially as clinical software needs to be protected against errors from changes introduced at a later point.

Future development needs to address the systematic support of API-enabled devices. Currently, a Telemed System (TELEMED Ltd., Vilnius, Lithuania) that offers full control over the ultrasound functionality is being integrated. Additionally, several diagnostic approaches that can serve as building bricks of new applications should be implemented. In particular, 3D segmentation [15] is an interesting topic in this context. MITK-US offers basic calibration support [10], but further development needs to be done in this direction as well.

From the point of view of clinical usage, there are several possible applications for navigated ultrasound. MITK-US served as an effective starting point for the development of the navigation solution. Tests have shown that the solution performs well and navigation is accurate [10]. The presented application is focused on needle insertion, but any tool can be tracked, which opens up other fields. Open liver surgery, for example, could profit from tracking scalpel and tumor to provide consistent results regarding the removal of tissue within a security margin around the tumor. Catheter-based minimally invasive approaches could also profit from tip-tracking of the catheter, especially as ultrasound is often used in intervention in the abdominal area.

It can be concluded that MITK-US achieves the development aims and that it is a viable development framework for US-based CAI applications in MITK. By using it, it is possible to quickly create and evaluate prototypes for ultrasound-based navigation systems. Should these systems prove to be viable, the certification process defined for MITK allows us to relatively quickly move the product into clinical trials. Its flexible nature together with the reusability of other MITK modules such as MITK-IGT makes us expect a high benefit for future projects.

References

1. Cleary K, Peters TM (2010) Image-guided interventions: technology review and clinical applications. Annu Rev Biomed Eng 12(1):119–142. doi:10.1146/annurev-bioeng-070909-105249. PMID: 20415592
2. Clevert DA, Paprottka PM, Helck A, Reiser M, Trumm CG (2012) Image fusion in the management of thermal tumor ablation of the liver. Clin Hemorheol Microcirc [Epub ahead of print]. doi:10.3233/CH-2012-1598
3. Figl M, Ede C, Hummel J, Wanschitz F, Ewers R, Bergmann H, Birkfellner W (2005) A fully automated calibration method for an optical see-through head-mounted operating microscope with variable zoom and focus. IEEE Trans Med Imaging 24(11):1492–1499
4. Franz AM, März K, Hummel J, Birkfellner W, Bendl R, Delorme S, Schlemmer H-P, Meinzer H-P, Maier-Hein L (2012) Electromagnetic tracking for US-guided interventions: standardized assess-

ment of a new compact field generator. Int J Comput Assist Radial Surg 7(6):813–818

5. Franz AM, Seitel A, Servatius M, Zöllner C, Gergel I, Wegner I, Neuhaus J, Zelzer S, Nolden M, Gaa J, Mercea P, Yung K, Sommer CM, Radeleff BA, Schlemmer HP, Kauczor HU, Meinzer HP, Maier-Hein L (2012) Simplified development of image-guided therapy software with MITK-IGT. In: Proceedings of SPIE medical imaging 2012: image processing, pp 83162J–83162J8. doi:10.1117/12.911421

6. Ince DC, Hatton L, Graham-Cumming J (2012) The case for open computer programs. Nature 482(7386):485–488. doi:10.1038/nature10836

7. Lasso A, Heffter T, Pinter C, Ungi T, Fichtinger G (2011) PLUS: An open-source toolkit for developing ultrasound-guided intervention systems. In: Proceedings 4th NCIGT NIH image guidedtherapy workshop, Vol 4, Arlington, VA, P 103

8. Maier-Hein L, Franz AM, Birkfellner W, Hummel J, Gergel I, Wegner I, Meinzer HP (2012) Standardized assessment of new electromagnetic field generators in an interventional radiology setting. Med Phys 39(6):3424–3434

9. Maier-Hein L, Tekbas A, Seitel A, Pianka F, Müller SA, Satzl S, Schawo S, Radeleff B, Franz RTAM, Müller-Stich BP, Wolf I, Kauczor HU, Schmied BM, Meinzer HP (2008) In vivo accuracy assessment of a needle-based navigation system for CT-guided radiofrequency ablation of the liver. Med Phys 35(12):5385–5396

10. März K, Franz AM, Stieltjes B, Iszatt J, Seitel A, Radeleff BA, Meinzer HP, Maier-Hein L (2013) Mobile em field generator for ultrasound guided navigated needle insertions. In: IPCAI, pp 71–80

11. Mercier L, Langø T, Lindseth F, Collins D (2005) A review of calibration techniques for freehand 3-d ultrasound systems. Ultrasound Med Biol 31(4):449–471. doi:10.1016/j.ultrasmedbio.2004.11.015

12. Müller M, Rassweiler MC, Klein J, Seitel A, Gondan M, Baumhauer M, Teber D, Rassweiler J, Meinzer HP, Maier-Hein L (2013) Mobile augmented reality for computer-assisted percutaneous nephrolithotomy. Int J Comput Assist Radiol Surg 8(4):663–675. doi:10.1007/s11548-013-0828-4

13. Muratore DM, Galloway RL (2001) Beam calibration without a phantom for creating a 3-d freehand ultrasound system. Ultrasound Med Biol 27(11):1557–1566

14. Navab N, Taylor R, Yang GZ (eds) (2012) Guest Editorial: Special issue on interventional imaging. IEEE Trans Med Imaging 12(4):857–859

15. Noble JA, Boukerroui D (2006) Ultrasound image segmentation: a survey. IEEE Trans Med Imaging 25(8):987–1010. doi:10.1109/TMI.2006.877092

16. Nolden M, Zelzer S, Seitel A, Wald D, Müller M, Franz A, Maleike D, Fangerau M, Baumhauer M, Maier-Hein L, Maier-Hein K, Meinzer HP, Wolf I (2013) The medical imaging interaction toolkit: challenges and advances. Int J Comput Assist Radiol Surg 8(4):607–620. doi:10.1007/s11548-013-0840-8

17. Read P, Meyer M (2000) Restoration of motion picture film. Butterworth–Heinemann series in conservation and museology. Elsevier, Amsterdam

18. Seitel A, Yung K, Mersmann S, Kilgus T, Groch A, Santos T, Franz A, Nolden M, Meinzer HP, Maier-Hein L (2012) MITK-ToF—range data within MITK. Int J Comput Assist Radiol Surg 7(1):87–96. doi:10.1007/s11548-011-0617-x

19. Teather R, Pavlovych A, Stuerzlinger W, MacKenzie I (2009) Effects of tracking technology, latency, and spatial jitter on object movement. In: IEEE symposium on 3D user interfaces, 2009 (3DUI 2009), pp 43–50. doi:10.1109/3DUI.2009.4811204

20. Wang DC, Chang W, Stetten GD (2005) Real-time ultrasound image analysis for the insight toolkit. Special issue of The Insight Journal, MICCAI 2005: Workshop on Open-Source Software

21. Wolf I, Vetter M, Wegner I, Böttger T, Nolden M, Schöbinger M, Hastenteufel M, Kunert T, Meinzer HP (2005) The medical imaging interaction toolkit. Med Image Anal 9(6):594–604. doi:10.1016/j.media.2005.04.005

22. Wu X, Taylor R (2003) A framework for calibration of electromagnetic surgical navigation system. In: Proceedings of the 2003 IEEE/RSJ international conference on intelligent robots and systems, 2003 (IROS 2003), vol 1, pp 547–552. doi:10.1109/IROS.2003.1250686