Published in final edited form as:

*Int J Comput Assist Radiol Surg*. 2012 January ; 7(1): 159–173. doi:10.1007/s11548-011-0636-7.

# TREK: an integrated system architecture for intraoperative cone-beam CT-guided surgery

**A. Uneri**,
Department of Computer Science, Johns Hopkins University, Traylor Building, Room #726, 720 Rutland Avenue, Baltimore, MD, 21205-2109, USA

**S. Schafer**,
Department of Biomedical Engineering, Johns Hopkins University, Traylor Building, Room #726, 720 Rutland Avenue, Baltimore, MD, 21205-2109, USA

**D. J. Mirota**,
Department of Computer Science, Johns Hopkins University, 224 NEB, 3400 N. Charles Street, NE Baltimore, MD, 21218, USA

**S. Nithiananthan**,
Department of Biomedical Engineering, Johns Hopkins University, Traylor Building, Room #726, 720 Rutland Avenue, Baltimore, MD, 21205-2109, USA

**Y. Otake**,
Department of Computer Science, Johns Hopkins University, 128 Hackerman Hall, 3400 N. Charles Street, NE Baltimore, MD, 21218, USA

**R. H. Taylor**,
Department of Computer Science, Johns Hopkins University, 127 Hackerman Hall, 3400 N. Charles Street, NE Baltimore, MD, 21218, USA

**G. L. Gallia**,
Department of Neurosurgery, The Johns Hopkins Hospital, Phipps Building, Room 118, 600 N. Wolfe Street, Baltimore, MD, 21287, USA

**A. J. Khanna**
Department of Orthopaedic Surgery, Johns Hopkins University, Good Samaritan Hospital, 5601 Loch Raven Blvd., Room G-1, Baltimore, MD, 21239, USA

**S. Lee**, **D. D. Reh**
Department of Otolaryngology, Head and Neck Surgery, Division of Rhinology and Sinus Surgery, Johns Hopkins University, Johns Hopkins University JHOC, 6th Floor, 601 N. Caroline Street, Baltimore, MD, 21287-0910, USA

**J. H. Siewerdsen**
Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

✉ jeff.siewerdsen@jhu.edu .

Conflict of interest

There is no conflict of interest beyond that stated in the Acknowledgments section - support from the National Institutes of Health and academic-industry partnership with Siemens Healthcare (Erlangen, Germany).

Department of Biomedical Engineering, Johns Hopkins University, Traylor Building, Room #718, 720 Rutland Avenue, Baltimore, MD, 21205-2109, USA

## Abstract

**Purpose—**A system architecture has been developed for integration of intraoperative 3D imaging [viz., mobile C-arm cone-beam CT (CBCT)] with surgical navigation (e.g., trackers, endoscopy, and preoperative image and planning data). The goal of this paper is to describe the architecture and its handling of a broad variety of data sources in modular tool development for streamlined use of CBCT guidance in application-specific surgical scenarios.

**Methods—**The architecture builds on two proven open-source software packages, namely the cisst package (Johns Hopkins University, Baltimore, MD) and 3D Slicer (Brigham and Women's Hospital, Boston, MA), and combines data sources common to image-guided procedures with intraoperative 3D imaging. Integration at the software component level is achieved through language bindings to a scripting language (Python) and an object-oriented approach to abstract and simplify the use of devices with varying characteristics. The platform aims to minimize offline data processing and to expose quantitative tools that analyze and communicate factors of geometric precision online. Modular tools are defined to accomplish specific surgical tasks, demonstrated in three clinical scenarios (temporal bone, skull base, and spine surgery) that involve a progressively increased level of complexity in toolset requirements.

**Results—**The resulting architecture (referred to as "TREK") hosts a collection of modules developed according to application-specific surgical tasks, emphasizing streamlined integration with intraoperative CBCT. These include multi-modality image display; 3D-3D rigid and deformable registration to bring preoperative image and planning data to the most up-to-date CBCT; 3D-2D registration of planning and image data to real-time fluoroscopy; infrared, electromagnetic, and video-based trackers used individually or in hybrid arrangements; augmented overlay of image and planning data in endoscopic or in-room video; and real-time "virtual fluoroscopy" computed from GPU-accelerated digitally reconstructed radiographs (DRRs). Application in three preclinical scenarios (temporal bone, skull base, and spine surgery) demonstrates the utility of the modular, task-specific approach in progressively complex tasks.

**Conclusions—**The design and development of a system architecture for image-guided surgery has been reported, demonstrating enhanced utilization of intraoperative CBCT in surgical applications with vastly different requirements. The system integrates C-arm CBCT with a broad variety of data sources in a modular fashion that streamlines the interface to application-specific tools, accommodates distinct workflow scenarios, and accelerates testing and translation of novel toolsets to clinical use. The modular architecture was shown to adapt to and satisfy the requirements of distinct surgical scenarios from a common code-base, leveraging software components arising from over a decade of effort within the imaging and computer-assisted interventions community.

### Keywords

Image-guided surgery; Cone-beam CT; Intraoperative imaging; Surgical navigation; System architecture; Open-source software

## Introduction

High-quality intraoperative imaging is recognized as essential to advancing the utility of surgical navigation across a broader spectrum of complex surgeries in which preoperative data alone is insufficient for guidance in the context of tissue deformation and excision. To this end, a prototype mobile C-arm for high-performance cone-beam CT (CBCT) has been developed [1], demonstrating sub-mm spatial resolution and soft-tissue visibility at low radiation dose. The system provides accurate information regarding the patient's anatomy beyond that of preoperative images and models, as is usually the case with traditional image guidance techniques, and allows reacquisition and update in a near-real-time manner to properly reflect anatomical change. Previous work using the prototype was found to improve performance in challenging surgical tasks [2] but relied solely on images (without navigational tools) and showed considerable room for improvement in workflow and surgical precision through integration of CBCT with tracking, registration with preoperative data, and improved visualization. This provides strong motivation to develop a streamlined, integrated guidance system in association with C-arm CBCT with a spectrum of application-specific capabilities. To accomplish this, a software architecture has been designed, featuring rigid and deformable 3D image registration, surgical trackers in single and hybrid arrangements, augmented reality methods using video and 3D model overlay, 3D-2D registration of preoperative data with real-time fluoroscopy, and graphical processing unit (GPU)-based digitally reconstructed radiographs (DRR). Although the software architecture was developed specifically for the C-arm CBCT system, it exercises a number of devices and toolkits (e.g., open-source software packages) and implements a variety of principles (e.g., modular binding of multiple data sources and complex numerical operations) that are pertinent to a broad scope of research and development efforts in image-guided surgery, particularly those involving system integration for intraoperative imaging.

While conventional surgical navigation approaches based upon preoperative images alone enjoy relative simplicity in system integration and registration (since the underlying image context is fixed), they suffer significant limitations with respect to anatomical change imparted prior to and during the procedure. The continually updated context of intraoperative CBCT creates a challenge in management of various guidance technologies, but also offers a means of streamlining manual procedures (e.g., manual point-based registration) that have hampered conventional approaches [3]. These, in combination with guidance tools adapted to specific surgical applications, help better utilize the continually updating CBCT image data to convey information to the operating surgeon. Recognizing the disparate needs and clinical tasks associated with various surgical applications—e.g. head-and-neck versus orthopaedic surgery—the architecture furthermore provides a flexible, modular platform for task-specific tool embodiments and workflow.

The last decade has seen vast development in the field of image-guided surgery and computer-assisted interventions. A number of commercial solutions have been translated to the operation room with varying degrees of success, including the StealthStation (*Medtronic*, Minneapolis, MN) and the Kolibri and VectorVision (*BrainLab*, Feldkirchen, Germany) systems, which provide streamlined navigation functionality. Such systems were originally conceptualized in the context of navigation relative to preoperative image data and have,

more recently, been demonstrated in combination with intraoperative imagers such as the O-arm (*Medtronic*, Minneapolis, MN) and intraoperative CT and MRI [4–6]. These developments reflect the opportunity and increasing demand for use of intraoperative imaging beyond conventional reliance on preoperative images alone.

In parallel with such commercial products, academic research has actively pursued the development of technologies that can support improved integration, better workflow, and incorporation of new technologies. A variety of software packages have been developed to answer different aspects of computer-assisted interventions [7–11]. An example open-source software toolkit is IGSTK [7] (*Georgetown University*, Washington, DC), which aims to simplify and accelerate the development of image-guided tracking and navigation systems, with emphasis on safety and robustness through its state machine architecture. A complementary, parallel development is represented by the cisst package [8] (*Johns Hopkins University*, Baltimore, MD) for interface to and integration of surgical trackers, video sources, and surgical robots. 3D Slicer focuses on visualization and image analysis [9] (*Brigham and Women's Hospital*, Boston, MA), sharing the same underlying VTK [12] and ITK [13] (*Kit-ware*, Clifton Park NY) toolkits as IGSTK [7] and MITK [10]. With additions like the OpenIGTLink communication protocol [14], it is compatible with devices such as surgical trackers and cameras via network link to an external application that supports such devices. This network layer presents a potential drawback with possible performance bottlenecks and latency (e.g., in handling streaming stereo HD video), and such issues remain areas of active development.

The purpose of this paper is to describe a software architecture (referred to as "TREK," connoting a general theme but not an acronym) motivated by the need for streamlined integration of C-arm CBCT with surgical navigation through the utilization of two popular open-source packages (the cisst package [8] and 3D Slicer [9]). The resulting platform leverages the strengths of each in realizing modular implementations for a variety of surgical applications centered on the use of C-arm CBCT. TREK is not intended as another toolkit, but rather is an application of open-source libraries that have grown from a decade of development throughout the imaging communities, instantiated in a form that emphasizes a number of novel aspects: (1) ease of development and rapid code prototyping through high-level language bindings of its underlying components, (2) a modular architecture allowing rapid assembly of task-specific implementations for a variety of surgical applications (e.g., modules for multi-modality tracking, image registration, and video data handling), and (3) an emphasis on real-time use within a dynamic context of continually updated intraoperative images (viz., cone-beam CT). The work is presented with two potential readerships within the computer-assisted surgery community—computer scientists and clinicians—with technical details of software architecture motivated by clinical challenges in CBCT guidance and example application-specific embodiments presented in a manner that is hopefully of interest to each. The "Materials" section below describes the variety of data sources associated with CBCT-guided surgical applications, and the "Methods" section details the software architecture developed to support modular navigation functionalities. In the "Result" section, specific embodiments of the architecture are reported in the context of three clinically relevant scenarios (deployed in a preclinical research environment) that make use of these modular toolsets in a progressively complex manner, ranging from fairly simple

functions (e.g., rigid registration of surgical planning data to CBCT) to more sophisticated operations (e.g., deformable registration and video augmentation). The work concludes with a discussion of the results and the future directions in architecture development for accelerating the translation of novel technologies to clinical use.

## Materials: data sources

A number of data sources are utilized in the surgical guidance system under development (Fig. 1), some of which are common to a modern operating room (OR), such as infrared trackers or conventional endoscopes, and others representing novel devices or implementations, as with video-based trackers, the tracked endoscope or even the intraoperative C-arm CBCT. The more common, commercially available devices and data sources were integrated based on manufacturers' specifications and application programming interfaces (APIs). The increased challenge of integrating with commercial, clinical systems involving their own (possibly proprietary) communication protocols is clearly recognized and points to ongoing efforts within the computer-assisted surgery and standards communities. While all the data sources illustrated in Fig. 1 are not necessarily used simultaneously, each plays a role in application-specific configurations of the system, where for example, one tracker embodiment may be better suited than another, or the application may call for a hybrid combination of data sources. The sections below briefly summarize the spectrum of data sources incorporated within the TREK software architecture.

### Prototype C-arm for cone-beam CT

The need for high-quality intraoperative 3D imaging has motivated the development of a mobile C-arm for CBCT. The prototype C-arm in Fig. 1 was developed in academic-industry partnership over the last several years [1], modifying an isocentric C-arm platform (PowerMobil, *Siemens Healthcare*, Erlangen, Germany) to include a flat-panel detector (PaxScan 3030 CB, *Varian Imaging Products*, Palo Alto, CA) in the imaging chain, a motorized orbit, and a system for geometric calibration and high-quality 3D image reconstruction. The device is capable of both 2D fluoroscopy and 3D CBCT, with the former performed as high as 30 fps and CBCT projection data for the latter typically at 3.3 fps, acquiring ~200 projections over 178°. CBCT imaging in applications such as skull-base and orthopaedic surgery [1,15–17] is fairly robust against artifacts arising from involuntary patient motion, while imaging in the presence of strong motion (e.g., cardiac motion) requires advanced acquisition and reconstruction methods—e.g., retrospective gating and motion-compensated reconstruction. Volume images demonstrate sub-millimeter spatial resolution and high-quality 3D visualization of bone and soft-tissues at low radiation dose [18]. By providing 3D image updates on demand, intra-operative CBCT can improve surgical navigation in a manner that properly accounts for anatomical deformation and excised tissue, thereby extending the role of image-guided surgery systems from rigid bony anatomy to a broader spectrum of surgical interventions. As illustrated in Fig. 1, the continually updated context of CBCT images lays the foundation for integration of various tracking and navigation technologies.

### Preoperative and intraoperative imaging

Depending on the disease and mode of therapeutic intervention, a variety of preoperative imaging modalities (e.g., CT, MR, and PET) may be used in delineating surgical targets and localizing adjacent critical normal anatomy. The value of intraoperative CBCT is significantly enhanced if such preoperative image data can be accurately registered to the most up-to-date image. Specifically, the CBCT image may be used to drive a deformable registration of preoperative information to an accurate, up-to-date context, providing the surgeon with superposition of multi-modality image and planning information; for example, deformable 3D-3D registration based on the Demons algorithm, allowing registration of CBCT to prior CBCT and preoperative CT (and, consequently, any information such MR, PET, and planning data registered to the preoperative CT [19]) can be implemented. In addition to integrating preoperative data sources with intraoperative CBCT, further data sources need to be accommodated from other intraoperative imaging, e.g. C-arm fluoroscopy relating the 3D scene through 3D to 2D registration techniques [20].

### Surgical planning

Registration of planning structures (usually defined in the context of preoperative images) provides an important aspect of surgical navigation. Typically, planning data (e.g., definition of surgical trajectories and cutlines as well as segmentation of anatomical landmarks, surgical targets, and normal anatomy) are registered to intraoperative CBCT and fluoroscopy using 3D-3D or 3D-2D image-image registration. In preclinical studies, external applications such as ITK-SNAP [21] (*University of Pennsylvania*, Philadelphia, PA) were used to define such planning structures either in preoperative CT or directly in intraoperative CBCT.

### Surgical trackers

Surgical trackers, such as stereoscopic video-based (T1 in Fig. 1), infrared (T2, T3), electromagnetic (T4), and mechanical trackers, are becoming prevalent in the modern OR, particularly in minimally invasive procedures in which the physical location of a tool must be precisely known in real-time relative to surrounding anatomy. The specific advantages and disadvantages of each define their usability in differing applications—e.g., tracking of rigid external tools with infrared trackers versus flexible internal instruments with an electromagnetic tracker—suggesting a role for multi-tracker support in next-generation guidance systems. The device APIs are accessed over various communication interfaces, e.g. serial (RS-232), USB and firewire (IEEE 1394), and make it possible to program and configure individual trackers for specific applications. Common usage requires a tracker to be registered to a tracked reference, which is often rigidly attached to the patient and positioned as to have optimal view of the reference marker and the target area.

### Video sources

Surgical video sources include endoscopic cameras (e.g., sinus endoscopes, and thoracosopes) as well as in-room video cameras, providing real-time images of the surgical scene through the use of mono and stereo cameras in conventional and high-definition formats. Additionally, video-based trackers (T1, V1 in Fig. 1) present potentially useful sources of video data. Projection radiographs synthesized from volumetric data sources

(DRRs) [22] can be incorporated as a video stream input similar to a camera as well. Near real-time streaming of these images with minimal lag is important to avoid latency and discrepancy with the physical world. Two video sources specifically considered below are a tracked sinus endoscope modified with infrared retroreflective markers (V2 in Fig. 1) and a video-based tracker (V1) mounted directly on the C-arm [23] for which the video feed is simultaneously used for tool tracking and an augmented video scene of the surgical field.

## Methods: software architecture

### Software foundation

The software architecture is primarily based on two open-source packages (cisst and 3D Slicer), which themselves utilize other open-source toolkits such as VTK [12], ITK [13], and OpenCV [24], thereby leveraging a broad base of expertise among developers and users. The two software packages complement each other in a manner that spans the requirements of the target applications, forming the TREK foundation as illustrated at a high level of abstraction in Fig. 2, with specific components of interest included in cisst and Slicer as detailed below.

**The cisst package—**The cisst package (Johns Hopkins University, Baltimore, MD) consists of software libraries designed to ease the development of computer-assisted intervention systems through a combination of core functionalities offering real-time performance in reading and/or controlling external devices [8]. It operates within a well-defined scope (i.e., separate from image visualization or processing) emphasizing a number of core procedures like device support, multitasking, and high-performance vector manipulation and is thus a useful complement to a medical image visualization solution (e.g., Slicer). The core cisst procedures as implemented in TREK are summarized below.

**Multi-task Framework:** *cisstMultiTask* provides the framework for creating and running tasks in parallel. Tasks can be defined to be event-based, continuous, or running with specific periodicities. They are connected to each other using a command pattern that specifies the provided/required interfaces for communication. Parallelism is achieved through both a shared memory model using threads and in a distributed manner using separate processes communicating over a local network. Tasks in the TREK platform use the shared memory approach, where input data can be propagated within the system with minimal memory copying.

**Device Support:** Attached to this multi-task framework, the *cisstDevices* library provides the layer that facilitates the device handling of a number of input sources. Surgical tracker support has been redesigned and improved in the course of TREK platform development, and support for new trackers was added to better handle manufacturers' APIs and communication protocols and to create similar-behaving interfaces to which other tasks may attach.

**Vision Pipeline:** The *cisstStereoVision* library, also used in conjunction with the multi-task framework, is used to generate the image processing pipeline that attaches to video sources. Tasks are broken down and defined in terms of filters that can execute over multiple threads

in parallel and, by utilizing the OpenCV package [24], achieve tasks such as automated calibration, color adjustment, and distortion correction.

**Transformation Manager:** Although alternatives exist in other packages used within the TREK architecture, the tight coupling to the physical data sources and the level of control it exposes motivated the use of *prmTransformationManager*. Changes in the periodically updating sources (e.g., trackers) are automatically reflected in this tree manager through the use of events. The manager allows support for error margins, which can be used to readily compute propagation of error. Support for representation of 2D and non-rigid transformation types is under development, which can extend its utility in our specific implementation of a multi-modality image guidance system.

**Data Collection Tools:** The *mtsCollector* type objects are attached to running tasks and collect the state of a given task and the events received by the task, where the output data may be generated in human-readable text or compressed binary format for immediate view, offline processing, or simple playback.

**Numerical Methods:** The *cisstVector* and *cisstNumerical* libraries are used at the core of a number of routines, either to store images or to carry out operations on transformations, thus enabling high computational efficiency through the use of stack-allocated storage and by replacing loop mechanisms with templated engines defined using recursive template metaprogramming.

**3D Slicer—**3D Slicer (Brigham and Women's Hospital, Boston MA) is a free, open-source software package for visualization and medical image analysis [9]. Its use in image-guided therapy is well established over the last decade through the functionalities provided in 3D visualization of multi-modality image data, advanced image analysis algorithms, segmentation, and therapy planning. Many of these capabilities form the basis of intraoperative image-guided procedures. A selection of these components is employed within the TREK architecture, as detailed below, in a manner that complements the cisst components detailed above. Together, the two packages provide a solid foundation for real-time, multitasked device control (cisst) combined with high-quality image visualization and analysis (Slicer).

**GUI Toolkit:** Employed as an end-user application, Slicer includes a cross-platform graphical user interface (GUI) toolkit, namely KWWidgets [25]. Widgets of specific interest are the module panel that contains the controls of various tools, the slice viewers for volumetric images, and 3D viewers that host the virtual scene. The 2D/3D viewers are wrapped VTK renderers forming the basis of most of the visualization methods available in TREK. The module panel is used to implement controls for various components of the system. For example, a collapsible style is used for the experimental prototype in basic laboratory studies, allowing simultaneous access to different portions of the system; on the other hand, the notebook style is used to provide simplified wizard-like interfaces for the clinical prototype exposing more application-specific workflow. The Module/View/Controller (MVC) architecture is also used to keep the logical implementation separate and to communicate with the interface and scene manager through events/listeners.

**Data Module (Volume Loader):** Slicer provides support for loading of images of various types and formats, including DICOM and MetaImage as the most common in our application (specifically, DICOM formatted CBCT and preoperative image data combined with MetaImage formatted planning structures). Multiple volumes/images can be loaded, rendered, and visualized concurrently, using 2D orthogonal/oblique planar views as well as various 3D view renderings.

**Editor Module (Surgical Planner):** Label maps are used to delineate planning data on the scan, which are then displayed in a superimposed manner on the slice views. Although primarily a preoperative procedure, having this capability in the OR allows on-the-fly editing of planning data, especially valuable with frequent intraoperative CBCT image updates. In practice, TREK accommodates a combination of preoperative planning in ITK-Snap [21] and/or intraoperative modifications via the Slicer Editor.

**MRML Scene:** The XML-based Medical Reality Modeling Language (MRML) is adopted through Slicer, which combines various types of data, e.g. volumes, images, fiducials, and planning, into a virtual representation of the surgical scene. This also provides a means to save or load a scene, allowing it to be recalled at a later point in the procedure without interrupting the normal course of an intervention (or experiment).

## TREK architecture for intraoperative image guidance

Built on the foundation described above and receiving a continuous flow of input from the data sources listed in the previous section, the TREK architecture is illustrated at a coarse level of package bindings illustrated in Fig. 2 and at a finer level of specific components in Fig. 3. As detailed below, TREK is best described in terms of the design decisions leading to streamlined integration of these software components in a manner that emphasizes modularity, execution speed, and ultimately the ability to build application-specific workflows that improve surgical performance.

**Scripting languages for integration of software packages—**The integration of multiple hardware devices and software packages traditionally comes at the cost of added complexity [26]. The TREK architecture seeks to tackle this problem through higher-level programming and the use of scripting to “glue” underlying components together. Individual packages, like cisst, Slicer and its underlying VTK and ITK toolkits are written in lower-level system programming languages (namely C/C++ or Fortran), offering data structures and algorithms that are often built from scratch from primitive computational elements and optimized to machine code through a compilation step. While this produces a highly efficient codebase, frequent compilation is often time-consuming, and the lower-level control provided by the language can overshadow the logical flow of an algorithm, hindering the development process through increasing complexity [27]. Compared with higher-level scripting languages like MATLAB (The Mathworks, Natick, MA) or Python (www.python.org), system languages are less expressive, require more lines of code, and the syntax is generally more complex due to exposing low-level control of memory allocation and manipulation. These problems are exacerbated when working with multiple packages, each with their own data types and definitions, and requiring intricate conversion

mechanisms for effective communication between each other. The solution adopted in TREK is to use a scripting language (Python) for simplified access to the various required packages, thereby combining solutions with unique standards and conventions under a common roof. Python, an interpreted, general-purpose, high-level programming language, was chosen to fulfill this task, emphasizing code readability, an extensive standard library with scientific solutions like NumPy (www.numpy.org) and SciPy (www.scipy.org), and existing bindings for many of the major packages used within TREK. As illustrated in Fig. 2, the TREK architecture utilizes Python bindings of cisst and Slicer packages to access the underlying components of each. This approach breaks the traditional build/debug cycle, since once the dependent components are compiled, the higher-level logic can be programmed without recompiling. Furthermore, as an alternative to conventional debugging, the current state of a program with all its variables can be observed through the (Python) interpreter in real time.

Communication between the scripted and compiled portions of the program takes place through an intermediary layer, called the language binding/wrapper. While the use of this additional layer poses a potential performance bottleneck (specifically with cisst due to its central role in device support structures), latency was minimized through the implementation of typemaps. Standard types like integers and floating-point numbers are seamlessly converted to their Python equivalents by an external application SWIG [26], which simply creates Python representations of these objects in memory. Other more custom types (e.g., Cartesian frames, color images, or image volumes – all vector-based constructs) are represented using NumPy, an extension to Python that adds support for multidimensional arrays and matrices along with a large library of high-level mathematical functions. To handle the conversion of such custom objects, typemaps have been designed to map the memory layout of a *cisstVector* to a NumPy array and back, thereby rendering physical memory copy unnecessary. As an analogous comparison, consider the MATLAB assignment operator, which creates a copy of the object with each new assignment, regardless of its type or size. The use of typemaps also enables extension of numerical methods in cisst (e.g., *cisstNumerical*), since mixing and matching of NumPy vectors with *cisstVectors* comes at minimal computational cost. Such implementations reduce latency that would otherwise arise from copy operations.

**Object-oriented programming for abstraction—**Control software for most of the data sources presented above introduces hardware dependencies. Most of the data sources include their own API and can communicate with the rest of the system using a specified communication protocol. TREK uses object-oriented programming to abstract these differences, simplify their control, and allow for easy extraction and management of information content. Figure 3 shows a detailed diagram of the TREK architecture emphasizing the abstraction of data sources through the use of cisst, Slicer, and language bindings. The hierarchical object structure formed in this example creates the two main objects of the TREK system, namely trekFrame and trekVideo.

As illustrated in Fig. 3, the input sources are intercepted on the compiled-side, with cisst handling the trackers and video sources, and Slicer managing the input of image data. Execution of these tasks is controlled from the scripted-side, either through user input or

from a preset configuration. The resolved data are then forwarded to the scripted-side to be represented in terms of two basic types: trekFrame represents objects in the scene that have a Cartesian position, whereas trekVideo is used for 2D images/video. Volumes, on the other hand, are handled natively by Slicer and are attached to trek-Frames. All objects are linked to VTK pipelines for rendering in the virtual scene, and their updates are handled behind the scene, triggered by the availability of new data. The objects also report back to the main packages—i.e., trekFrame and trekVideo are attached to the cisst transformation manager and vision pipeline, whereas Slicer perceives them as MRML transform and camera nodes, respectively, thus ensuring compatibility with other functionalities provided by these foundation packages (e.g., existing Slicer plugins).

**Build system and runtime configuration—**The component-based design of TREK follows all the way down to its build system. The open-source build system CMake was chosen to handle the numerous software packages comprising TREK. The experimental CMake-based build system of Slicer version 4 was backported and modified to include automatic compilation of all packages required by TREK. Additional functionality was implemented to automate new module creation from templates.

Although most parameters of a running TREK session can be configured interactively, it was observed that using configuration files greatly saved time during an experiment. TREK configuration files are used to initialize and configure the system to adapt to specific applications. Extensible markup language (XML) file format is used to describe the inputs and various parameters of the system. Upon load, the files are validated against a Document Type Definition (DTD), followed by a sanity testing of the values. This way, an application-specific set of modules, navigation tools, preferences, and workflow can be launched based on a single configuration file.

**Registration of data sources—**The technical integration of data sources with the software components that abstract and drive them facilitates a final stage of algorithmic integration in which volumes, images, and coordinate frames are related to each other by means of registration. Although both cisst and Slicer offer some readymade solutions for tracking and registration, there are numerous aspects of CBCT imaging (e.g., truncation, artifacts, and CT number inaccuracy) and, more importantly, a number of advanced registration techniques under development (e.g., automatic image-world registration [3], image-image registration [19], and image-video registration [28]) that motivate a higher degree of control and flexibility. The flexibility of the TREK architecture in this regard is put to test in accommodating the following registration tasks essential for 3D-3D and 3D-2D image-image registration, image-world (tracker) registration, and image-video registration.

**Image Registration—**Methods of image registration implemented in TREK include simple point-based methods, image-based rigid registration, and image-based deformable registration methods (e.g., a multi-scale variant of the Demons algorithm [19]). The primary toolkit underlying such algorithms is ITK, although a variety of algorithms represent areas of continual development and future research.

**Tracker Registration—**The nominal method for image-world (tracker) registration involves rigid point co-localization of fiducials, relying on markers visible in both the patient image and the tracker coordinate frame. Horn's method for rigid registration was implemented in NumPy for such manual localization [29]. An alternative registration method involves automatic localization of fiducials visible to both the C-arm X-ray imaging and the tracker—e.g., retroreflective markers with tungsten BBs at the center of each fiducial [3]. Another derivation of the problem solves the $AX = XB$ problem to register different trackers to one another [30].

**Video Registration—**In addition to handling various video data sources streamlining as trekVideo objects, registration of image and video data includes a conventional, tracker-based registration [31] of endoscopic video with CBCT as well as a more precise image-based registration [32] that further utilizes feature points detected in the video and CBCT scene. The former is fully integrated in TREK, using basic camera calibration utilities (e.g., DLR CalLab and CalDe [33]) for real-time registration and video overlay of CBCT data (and any other image or planning data registered to CBCT) to the video stream supplied by a tracked endoscope. A working prototype of the image-based approach is implemented in MATLAB, which can be executed from TREK using a customized version of mlabwrap library that allows its control through Python. Development is underway to refactor the computationally intensive portions of image segmentation and feature identification and translate them to more optimized library methods in cisst and OpenCV.

**3D-2D Registration—**Registration of 3D virtual scenes (e.g., CBCT images or 3D planning structures) to C-arm fluoroscopy involves highly parallelizable computations implemented in CUDA to allow GPU-accelerated forward-projection methods (e.g., variations on the Siddon algorithm) [34]. TREK calls upon a custom toolkit of such CUDA functions, allowing input pose determination from within the GUI or from a tracker (e.g., a trekFrame object linked to a tracked, handheld tool). For 3D-2D registration, TREK provides the means for computing such fast forward-projection 2D images (e.g., DRRs computed on preoperative CT, planning data, and/or intraoperative CBCT) with the fluoroscopic image by means of various iterative methods (e.g., 2D image gradient information similarity metrics [35] with a covariance matrix adaptation evolution [36]).

**Online analysis of geometric uncertainty—**TREK includes the ability to perform fairly extensive online analysis of geometric accuracy, as opposed to a conventional approach of recording data for subsequent offline analysis. Registration errors are computed and reported for fiducials (FRE) and targets (TRE), which can be arbitrarily defined [37]. Having such analysis exposed directly in the navigation interface helps to guide selection of fiducials in a manner that minimizes registration errors. Such is also consistent with a progressive philosophy of better communicating sources of geometric error to the surgeon [38], helping to overcome the current qualitative, mental balancing act that a surgeon needs to perform in factoring what the navigation system is showing (e.g., suggesting "perfect" precision) against what s/he knows to be likely sources of error. An example involves 3D colormaps of TRE ($x, y, z$) that can be generated and overlaid on the navigation scene to better convey the fidelity of tool localization based on the spatial distribution of registration

fiducials [39], or similarly a 3D “cloud” or “cone” with diameter proportional to TRE superimposed on tracked tools to communicate the uncertainty in tooltip localization [38]. Similarly, TRE could be fused with the endoscopic scene in video-CT registration [40,28]. Such tools are a possibly important method of conveying tolerance and geometric error to the surgeon, and their effect on surgical performance and decision-making is the subject of ongoing work. Aside from this more clinically oriented motivation for online analysis, such tools are valuable in rigorous laboratory performance assessment. The same analysis tools and numerical methods coupled with a Python scripting interface yield a powerful platform for experimentation, capable of not just collecting and distributing data, but also processing and visualizing results in real-time.

## Results

The resulting architecture facilitates highly application-specific implementations that are well suited to specific surgical tasks and workflow scenarios through adaptable combinations of the modular components described above. Three example applications are demonstrated below, representing an increasing level of complexity with respect to three specific surgical applications. In each case, we describe the technical implementation of TREK modules, followed by demonstration in a specific clinical scenario.

### Display and registration of CBCT, planning, and preoperative image data

**Technical implementation—**The fundamental task implicit to all applications of TREK is the display of up-to-date CBCT images. A possible exception is a scenario in which CBCT is only used as a framework (under the hood) to drive the registration of other-modality preoperative images to an up-to-date context, but in all applications considered below, direct visualization of CBCT is basic to the image guidance system. Slicer natively handles loading and display of such image data via DICOM import of CBCT images. Triplanar 2D slice view renderers allow superimposing of CBCT slices with registered preoperative data to provide a more complete perspective on surrounding anatomical context while displaying the most up-to-date CBCT images within the region of interest about the surgical target. The field of view boundary can also communicate the accuracy of image registration via the continuity of structures extending between the CBCT and preoperative image. Using layer transparency, fading, or difference images between successive scans provides quick assessment of tissue changes—e.g. excision or deformation.

The surgical plan is superimposed in triplanar views, and for 3D views, rendered as smoothed triangular meshes. High-contrast bone is also typically rendered to provide greater context (Fig. 4) obtained by thresholding a reduced version of the preoperative image, and generating a decimated triangular mesh. Finally, 3D-3D image registration may be invoked by way of simple point-based or image-based rigid registration or more sophisticated deformable registration. TREK invokes standard ITK tools for rigid image registration. The deformable Demons algorithm described in section “Registration of data sources” has been fairly rigorously tested in cadaveric and clinical data, requiring ~10–20s for registration of $256^3$ voxel CBCT volumes (trekDeformable-Registration). Its use enables the flow of

information from preoperative imaging and planning to intraoperative imaging and surgical delivery.

**Clinical example: temporal bone surgery—**A specific embodiment of the TREK architecture is illustrated in Fig. 4 for guidance of temporal bone surgery. The anatomy of the temporal bone includes numerous mm-scale features in complex 3D orientations that can be difficult to visualize in 2D triplanar views. Displaying anatomical landmarks, the surgical target, and critical boundaries within both cross-sectional and fully 3D representations augments the surgeon's mental map. In cochlear implantation, for example, C-arm CBCT has been identified as a potentially useful guidance modality (particularly in cases of abnormal anatomy) [15], with high-resolution CT providing useful preoperative planning.

Conventional approaches leverage real-time tracking relative to preoperative CT, possible incorporation of complex fixation/immobilization devices, and a fair amount of "mental registration" on the part of the surgeon to reconcile disparity between the preoperative image data and the actual anatomy during the course of intervention. The TREK embodiment illustrated in Fig. 4 has the potential to bypass many of these shortcomings through use of intraoperative CBCT complemented by high-resolution preoperative CT and surgical planning. The updated 3D image context serves to properly display anatomical changes (e.g., excision of the mastoid bone) and account for tissue deformation (e.g., shift of the brain and dura following the cerebrospinal fluid drainage). Furthermore, the up-to-date CBCT images provide freedom from restrictive registration processes and immobilization devices, so long as the surgical target is within the CBCT field of view (FOV).

## Tool tracking and augmented video

**Technical implementation—**TREK incorporates single/multiple surgical trackers to localize the position and orientation of surgical instruments ranging from simple pointers to drills, suction, and tracked endoscopes. The architecture accommodates simultaneous capturing from the spectrum of tracking modalities through its underlying cisstMultiTask framework. The trackers are managed from the Python side (trekTrackers), where they are instantiated as separate C threads to run periodically at 1 kHz. Once data are received from the external device, they are shared in memory along interfaces and over language bindings via SWIG typemaps, thus minimizing costly memory copy operations that could otherwise introduce lag. Each tracked tool is mapped to a trekFrame object, synchronizing data from the tracker automatically and relating to the rest of the transformation tree defined at configuration-time. The tools can be visualized in 3D or 2D views, either by centering the slices to the tooltip (Fig. 5a) or by moving the crosshairs within a fixed field of view when a greater context is required (Fig. 6a).

While intraoperative 3D imaging provides an up-to-date view of patient anatomy, the mental correspondence of the physical world to 3D images can be challenging, especially for procedures in which the surgeon operates through a small incision using a video endoscope that is free to move and rotate, but the perspective of which may bear little resemblance to triplanar or fully 3D views. In this respect, the ability to register image and planning data directly to the video endoscopic scene can provide a valuable window on a complex

set of 3D data within a natural visual scene. Leveraging similar components as for trackers (above), video inputs can be captured simultaneously in a multi-threaded manner. Due to the increased input data size, common operations such as adding alpha channels to video frames, resizing, and rectifying for distortion are carried out in thread pools for parallel processing. Aside from synchronizing image data and performing basic pipeline processing, a trekVideo component keeps track of camera intrinsic and extrinsic parameters and is attached to other trekFrames for correct positioning in the virtual scene. Using these functionalities, by switching the default scene camera with a trekVideo frame, the video image can be augmented with planning structures (trekVideoAugmentation) as illustrated in Fig. 5a in an alpha-blended fashion. Analysis of video registration accuracy shows ~2 mm target registration error (TRE) obtained from tracker-based video registration, improved to sub-mm accuracy with the video-CBCT image-based technique [28].

**Clinical example: skull base surgery—**A slightly more complex combination of TREK display and tracking modules is illustrated in Fig. 5 for application in endoscopic skull base surgery. Such intervention involves the excision of tumors located in and about the cribriform and clivus, where the surgical target is often surrounded by a complex 3D architecture of critical structures such as the carotid arteries and optic nerves. Up-to-date imaging that properly reflects anatomical change and allows confident visualization of the tumor and critical structures therefore offers a potentially significant advance to the state of the art. A total of 3–5 CBCT scans has been found in preclinical studies to guide the surgeon in the surgical approach, initial resection, further resection of residual tumor if necessary, and final image-based verification and quality assurance of the surgical product.

Figure 5 shows the case in which a cadaver is used in preclinical studies of surgical performance under CBCT guidance. The surgical tools common to such transnasal approaches make the use of optical trackers practical, since the position and orientation of the tools are naturally constrained by the anatomy, and line-of-sight issues can be minimized. To this end, an infrared (T3) and/or a video-based tracker (T1,V1) may be used to track the surgical tools, endoscope (V2), and the patient via a reference marker attached to a Mayfield clamp (Fig. 5b). Not only are preoperative images and planning data registered to the most up-to-date CBCT (as described in the previous section and shown in the triplanar views of Fig. 5a), but real-time tracking of the video endoscope allows registration of these data directly within the video scene. An alpha-blended rendering of planning structures within the video scene thus allows the skull base surgeon to see beyond the surface anatomy and more precisely approach and excise the surgical target while avoiding surrounding critical anatomy. Soft tissue visibility achieved by CBCT, coupled with deformable registration that properly accounts for effects such as mandibular motion, neck flexion, herniation of orbital contents, and excised sinus architecture allows for an up-to-date, geometrically resolved rendering with each scan.

## Virtual fluoroscopy and digitally reconstructed radiographs

**Technical implementation—**A third level of modular implementations aims to address opportunities for dose reduction (a topic of considerable interest to the clinical community) as well as high-precision 3D-2D registration. To this end, a DRR generating module

has been implemented (trekDRR) which interfaces to a custom-designed GPU toolkit implemented in CUDA. Tracked tools (trekFrame) are attached to generate DRRs from arbitrary poses, and due to the 2D nature of the synthesized images, the output of the module is treated as a video source, thus using the previously described solutions for display and processing (trekVideo). Embodiments of this functionality undergoing preclinical evaluation include (i) virtual fluoroscopy from the perspective of a handheld tool or the C-arm itself, (ii) 3D correspondence of a 2D image with respect to the surgical scene, and (iii) tools to assist C-arm tableside setup such that the C-arm orbit is free of collision and the target is within the CBCT FOV. Each of these offers the potential to reduce reliance on fluoroscopy for simple localization tasks and could provide more streamlined workflow scenarios, reduction in surgery time, and reduced radiation dose, as illustrated for the example of spine surgery below.

**Clinical example: orthopaedic spine surgery—**Orthopaedic surgery is among the fields that have most fully adopted and benefited from conventional image guidance approaches, demonstrating improvements in surgical outcome associated with such capability [41]. However, current embodiments also highlight the shortfalls of conventional approaches—not only the reliance on preoperative image data but also bottlenecks in workflow, a lack of system integration, and only basic attempts to communicate potential geometric errors to the surgeon. An assembly of TREK modules for spine surgery suggests a number of potential advances, such as automatic registration techniques [3] that allow image-world registration to be updated with each CBCT scan, removing one of the workflow bottlenecks and maintaining a higher level of geometric accuracy throughout the procedure. Although rigid registration of targets may be sufficient in some cases, incorporation of piecewise rigid (e.g., individual vertebrae) or fully deformable registration offers to better accommodate intraoperative change. One of the main goals of the prototype C-arm was to deliver 3D images with spatial and contrast resolution beyond that of existing CBCT systems, facilitating visualization of soft-tissues (e.g., muscle, fat, and tumor) throughout the FOV. As shown in Fig. 6a, superposition of CBCT within the larger FOV of preoperative images provides a useful large-scale context in which to interpret intraoperative imaging. Similarly, registration of planning structures and surgical trajectories (e.g., transpedicular approach to the vertebral body) offers enhanced visualization that could improve surgical precision and reduce surgery time (e.g., time spent searching for incision points and tool advancement in the pedicle). Radiation dose may be further reduced through the use of low-dose CBCT acquisition protocols [17] and also by using virtual fluoroscopy and 3D-2D registration to quickly localize anatomy without imparting any dose or reduce to a single confirmatory image—a process that conventionally requires multiple fluoroscopy shots and potentially prolonged exposure times (i.e., search/localization). Video augmentation (e.g., overlay of image and planning data within the tracker-on-C video stream) provides immediate, intuitive integration of information during tool advancement into the spine. The ability to obtain a high-quality CBCT image at or near the completion of a procedure, registered to planning data, offers a potentially valuable means of verifying no-tools-left-behind, evaluating the quality of surgical product and assuring other patient safety considerations.

## Discussion and conclusions

The development of a new software integration architecture for C-arm CBCT-guided interventions has been reported, addressing a variety of inherent limitations of conventional image guidance within preoperative data alone. Exploiting a modular architecture, the TREK system integrates tools that particularly leverage the advantages of intraoperative CBCT, while allowing for variations in implementation within the context of specific surgical applications and intraoperative imaging scenarios (e.g., other intraoperative imaging modalities). Specific embodiments of the architecture in surgery of the temporal bone, skull base, and spine were demonstrated, illustrating the potential to adapt to different clinical scenarios with varying requirements. The use of complementary open-source software packages (namely cisst and 3D Slicer) as the foundation of the architecture provides a toolset that allows the implementation of new functionalities that would otherwise not be possible by either alone.

The relative ease of integration of software packages through scripting allowed us to rapidly develop and iterate upon task-specific embodiments in preclinical studies, thereby accelerating their translation to clinical use. The video registration module presented in section “Registration of data sources” is a good example, where previously prototyped MATLAB scripts were integrated with the system through Python, thereby rendering simple, high-level programming tools useful alongside other modules. The subsequent refactoring process relegates computationally intensive portions of the code down to the low-level, optimized library level. This prototyping/refactoring cycle simplifies and speeds the development process, since even an initial version of a module employing high-level language integration can be integrated with the system in a manner that gives runtime access to intraoperative data. This is believed to reduce conventional barriers of translating from prototype to preclinical/clinical deployment. Although TREK and its modules are not planned for open-source release in themselves, the development has rendered some benefit and return to the open-source community in terms of bug fixes and enhancements to the open-source packages used—e.g., added support for a broad variety of trackers now handled within the cisst package.

Runtime configuration of the setup for specific applications is currently achieved through a configuration description that is parsed during startup. Ongoing development of the cisst package seeks to offer greater flexibility to this design through the use of dynamic connection of tasks, which would allow reconfiguring the system on-the-fly, providing flexibility in attaching/detaching tools and trackers and adding/toggling filters of the imaging pipelines. The success of GPU implementations of various computational methods in the TREK modules also motivated transfer of the integrated code to a custom toolkit, allowing better experimentation with novel methods of forward/back projection, reconstruction, and 3D registration methods. The toolkit is envisioned to be used from within TREK modules and dynamically linked to the rest of the architecture.

## Acknowledgments

This work was supported in part by the National Institutes of Health R01-CA-127444. The prototype C-arm was developed in academic-industry partnership with Siemens Healthcare (Erlangen, Germany). The authors extend

sincere thanks to Dr. Peter Kazanzides, Mr. Anton Deguet, Mr. Balazs Vagvolgyi, and Mr. Daniel Li (Department of Computer Science, Johns Hopkins University) for assistance with the cisst package. Preclinical studies were carried out in the Johns Hopkins Hospital Minimally Invasive Surgical Training Center (MISTC) with support from Dr. Michael Marohn and Ms. Sue Eller (Johns Hopkins Medical Institute).

## References

1. Siewerdsen JH, Moseley DJ, Burch S, Bisland SK, Bogaards A, Wilson BC, Jaffray DA (2005) Volume CT with a flat-panel detector on a mobile, isocentric C-arm: pre-clinical investigation in guidance of minimally invasive surgery. Med Phys 32(1):241–254 [PubMed: 15719975]
2. Chan Y, Siewerdsen JH, Rafferty MA, Moseley DJ, Jaffray DA, Irish JC (2008) Cone-beam computed tomography on a mobile C-arm: novel intraoperative imaging technology for guidance of head and neck surgery. J Otolaryngol Head Neck Surg 37(1):81–90 [PubMed: 18479633]
3. Hamming NM, Daly MJ, Irish JC, Siewerdsen JH (2009) Automatic image-to-world registration based on X-ray projections in cone-beam CT-guided interventions. Med Phys 36(5):1800–1812 [PubMed: 19544799]
4. Zhang J, Weir V, Fajardo L, Lin J, Hsiung H, Ritenour ER (2009) Dosimetric characterization of a cone-beam O-arm imaging system. J Xray Sci Technol 17(4):305–317 [PubMed: 19923687]
5. Lunsford LD, Parrish R, Albright L (1984) Intraoperative imaging with a therapeutic computed tomographic scanner. Neurosurgery 15(4):559–561 [PubMed: 6493465]
6. Jolesz FA (1998) Interventional and intraoperative MRI: a general overview of the field. J Magn Reson Imaging 8(1):3–7 [PubMed: 9500253]
7. Gary K, Ibanez L, Aylward S, Gobbi D, Blake MB, Cleary K (2006) IGSTK: an open source software toolkit for image-guided surgery. Computer 39(4):46–53
8. Deguet A, Kumar R, Taylor R, Kazanzides P (2008) The cisst libraries for computer assisted intervention systems. MIDAS J Syst Archit Comput Assist Interv
9. Pieper S, Lorensen B, Schroeder W, Kikinis R (2006) The NA-MIC Kit: ITK, VTK, pipelines, grids and 3D slicer as an open platform for the medical image computing community. In: Proceedings of IEEE International Symposium Biomedical Imaging, 6–9 April 2006. pp 698–701
10. Wolf I, Nolden M, Boettger T, Wegner I, Schoebinger M, Hastenteufel M, Heimann T, Meinzer H-P, Vetter M (2005) The MITK approach. Insight Journal—MICCAI Open-Source Workshop
11. Rexilius J, Spindler W, Jomier J, Koenig M, Hahn H, Link F, Peitgen H-O (2005) A framework for algorithm evaluation and clinical application prototyping using ITK. Insight Journal—MICCAI Open-Source Workshop
12. Schroeder W, Lorenson B (1996) Visualization toolkit: an object-oriented approach to 3-D graphics. Prentice Hall PTR, Englewood Cliffs
13. Ibanez L, Schroeder W, Ng L, Cates J (2005) The ITK software guide. MIDAS J Syst Archit Comput Assist Interv
14. Tokuda J, Fischer GS, Papademetris X, Yaniv Z, Ibanez L, Cheng P, Liu H, Blevins J, Arata J, Golby AJ, Kapur T, Pieper S, Burdette EC, Fichtinger G, Tempany CM, Hata N (2009) OpenIGTLink: an open network protocol for image-guided therapy environment. Int J Med Robot 5(4):423–434 [PubMed: 19621334]
15. Rafferty MA, Siewerdsen JH, Chan Y, Daly MJ, Moseley DJ, Jaffray DA, Irish JC (2006) Intraoperative cone-beam CT for guidance of temporal bone surgery. Otolaryngol Head Neck Surg 134(5):801–808 [PubMed: 16647538]
16. Daly MJ, Siewerdsen JH, Moseley DJ, Jaffray DA, Irish JC (2006) Intraoperative cone-beam CT for guidance of head and neck surgery: assessment of dose and image quality using a C-arm prototype. Med Phys 33(10):3767–3780 [PubMed: 17089842]
17. Schafer S, Nithananiathan S, Mirota DJ, Uneri A, Stayman JW, Zbijewski W, Schmidgunst C, Kleinszig G, Khanna AJ, Siewerdsen JH (2011) Mobile C-Arm cone-beam CT for guidance of spine surgery: image quality, radiation dose, and integration with interventional guidance. Med Phys (to appear)
18. Ritter D, Orman J, Schmidgunst C, Graumann R (2007) 3D Soft tissue imaging with a mobile C-arm. Comput Med Imaging Graph 31(2):91–102 [PubMed: 17188841]

19. Nithiananthan S, Brock KK, Daly MJ, Chan H, Irish JC, Siewerdsen JH (2009) Demons deformable registration for CBCT-guided procedures in the head and neck: convergence and accuracy. Med Phys 36(10):4755–4764 [PubMed: 19928106]
20. Munbodh R, Jaffray DA, Moseley DJ, Chen Z, Knisely JPS, Cathier P, Duncan JS (2006) Automated 2D-3D registration of a radiograph and a cone beam CT using line-segment enhancement. Med Phys 33(5):1398–1411 [PubMed: 16752576]
21. Yushkevich PA, Piven J, Hazlett HC, Smith RG, Ho S, Gee JC, Gerig G (2006) User-guided 3D active contour segmentation of anatomical structures: significantly improved efficiency and reliability. NeuroImage 31(3):1116–1128 [PubMed: 16545965]
22. Sherouse GW, Novins K, Chaney EL (1990) Computation of digitally reconstructed radiographs for use in radiotherapy treatment design. Int J Radiat Oncol Biol Phys 18(3):651–658 [PubMed: 2318699]
23. Reaungamornrat S, Otake Y, Uneri A, Schafer S, Stayman JW, Zbijewski W, Mirota DJ, Yoo J, Nithiananthan S, Khanna AJ, Taylor RH, Siewerdsen JH (2011) Tracker-on-C: a novel tracker configuration for image-guided therapy using a mobile C-arm. In: Computer Assisted Radiology and Surgery, Berlin, Germany, 22–25 June 2011. CARS (to appear)
24. Bradski G, Kaehler A (2008) Learning OpenCV: computer vision with the OpenCV library. O'Reilly Media, Sebastopol
25. Oguz I, Gerig G, Barre S, Styner M (2006) KWMeshVisu: a mesh visualization tool for shape analysis. Insight Journal—MICCAI Open-Source Workshop
26. Beazley DM (2003) Automated scientific software scripting with SWIG. Futur Gener Comput Syst 19(5):599–609
27. Ousterhout JK (1998) Scripting: higher level programming for the 21st Century. Computer 31(3):23–30
28. Mirota DJ, Uneri A, Schafer S, Nithiananthan S, Reh DD, Gallia GL, Taylor RH, Hager GD, Siewerdsen JH (2011) High-accuracy 3D image-based registration of endoscopic video to C-arm cone-beam CT for image-guided skull base surgery. In: Wong KH, Holmes Iii DR (eds) SPIE medical imaging: visualization, image-guided procedures, and modeling. SPIE, Lake Buena Vista, pp 79640J–79610
29. Horn BKP (1987) Closed-form solution of absolute orientation using unit quaternions. J Opt Soc Am A 4(4):629–642
30. Park FC, Martin BJ (1994) Robot sensor calibration: solving AX=XB on the Euclidean group. IEEE Trans Robot Autom 10(5):717–721
31. Daly MJ, Chan H, Prisman E, Vescan A, Nithiananthan S, Qiu J, Weersink R, Irish JC, Siewerdsen JH (2010) Fusion of intraoperative cone-beam CT and endoscopic video for image-guided procedures. In: Wong KH, Miga MI (eds) SPIE medical imaging: visualization, image-guided procedures, and modeling. SPIE, San Diego, pp 762503–762508
32. Mirota D, Wang H, Taylor R, Ishii M, Hager G (2009) Toward video-based navigation for endoscopic endonasal skull base surgery. In: Yang G-Z, Hawkes D, Rueckert D, Noble A, Taylor C (eds) Medical Image Computing and Computer Assisted Intervention. Lecture notes in computer science, vol 5761. Springer, Berlin, Heidelberg, pp 91–99
33. Strobl KH, Hirzinger G (2006) Optimal hand-eye calibration. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 9–15 Oct. 2006, pp 4647–4653
34. Siddon RL (1985) Fast calculation of the exact radiological path for a three-dimensional CT array. Med Phys 12(2):252–255 [PubMed: 4000088]
35. Pluim J, Maintz J, Viergever M (2000) Image registration by maximization of combined mutual information and gradient information. In: Delp S, DiGoia A, Jaramaz B (eds) Medical Image Computing and Computer Assisted Intervention. Lecture notes in computer science, vol 1935. Springer, Berlin, Heidelberg, pp 103–129
36. Hansen N (2006) The CMA evolution strategy: a comparing review. In: Lozano J, Larrañaga P, Inza I, Bengoetxea E (eds) Towards a new evolutionary computation, vol 192. Studies in fuzziness and soft computinge. Springer, Berlin, Heidelberg pp 75–102
37. Fitzpatrick JM, West JB, Maurer CR Jr (1998) Predicting error in rigid-body point-based registration. IEEE Trans Med Imaging 17(5):694–702 [PubMed: 9874293]

38. Simpson AL, Ma B, Ellis RE, Stewart AJ, Miga MI (2011) Uncertainty propagation and analysis of image-guided surgery. In: Wong KH, Holmes Iii DR (eds) SPIE medical imaging: visualization, image-guided procedures, and modeling. SPIE, Lake Buena Vista, Florida, USA pp 79640H–79647

39. Hamming NM, Daly MJ, Irish JC, Siewerdsen JH (2008) Effect of fiducial configuration on target registration error in intraoperative cone-beam CT guidance of head and neck surgery. In: Proceedings of the IEEE Engineering in Medicine and Biology Society, 20–25 Aug. 2008, pp 3643–3648

40. Higgins WE, Helferty JP, Lu K, Merritt SA, Rai L, Yu K-C (2008) 3D CT-video fusion for image-guided bronchoscopy. Comput Med Imaging Graph 32(3):159–173 [PubMed: 18096365]

41. DiGioia AM 3rd, Jaramaz B, Colgan BD (1998) Computer assisted orthopaedic surgery. Image guided and robotic assistive technologies. Clin Orthop Relat Res 354:8–16


**Fig. 1.**

Prototype C-arm for CBCT, tracking and navigation technologies. Systems labeled in the photograph include the C-arm, tracked tools, *T1* (MicronTracker, *Claron*, Toronto ON), *T2* (Polaris Spectra, *NDI*, Waterloo ON), *T3* (Polaris *Vicra*, *NDI*), *T4* (Aurora, *NDI*), *V1* (MicronTracker, *Claron*), *V2* (Video endoscope, 7230AA, *Karl Storz*, Tuttlingen Germany), *C1* (TREK navigation workstation), *C2* (C-arm image acquisition workstation), *D1* (TREK navigation displays), and *D2* (C-arm image displays). C2/D2 are traditionally separate from the tableside setup, operated by a technologist to acquire and process C-arm images and optionally exposed to the surgeon for image review. Instead, images from C2/D2 are acquired by the technologist and transferred to C1/D1 for real-time guidance and visualization by the surgeon

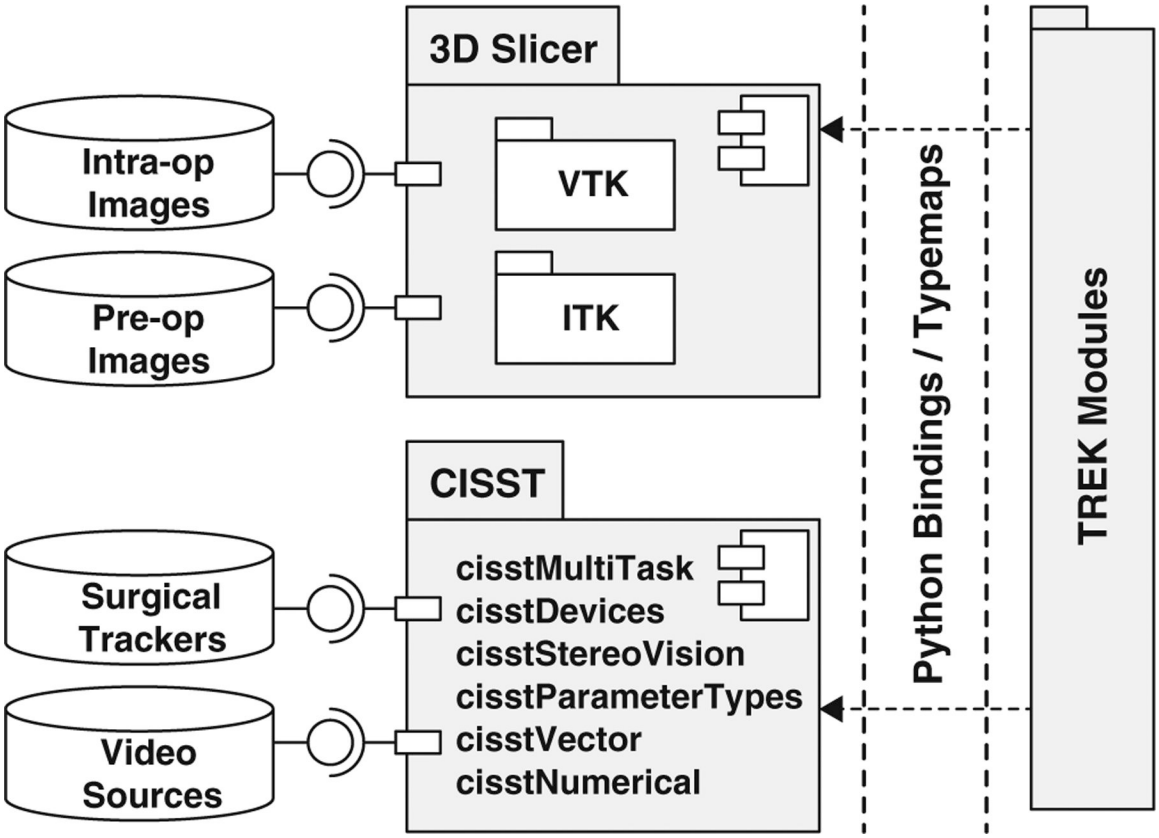


**Fig. 2.**

Component diagram illustrated using unified modeling language (UML) showing an overview of the TREK system architecture with input data sources, primary software internals, and binding layer to modular components

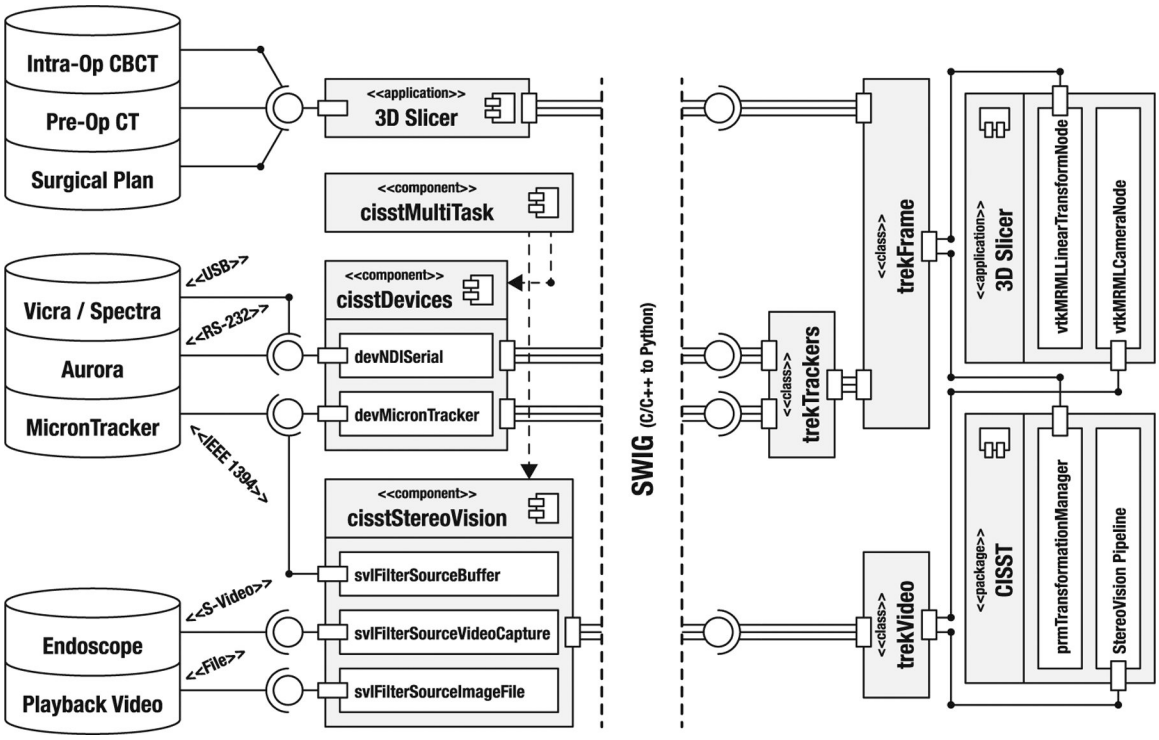


**Fig. 3.**

Illustrative abstraction of the TREK architecture, emphasizing the trekFrame and trekVideo objects. The left and right sides of the figure depict the compiled and scripted portions of the implementation, respectively

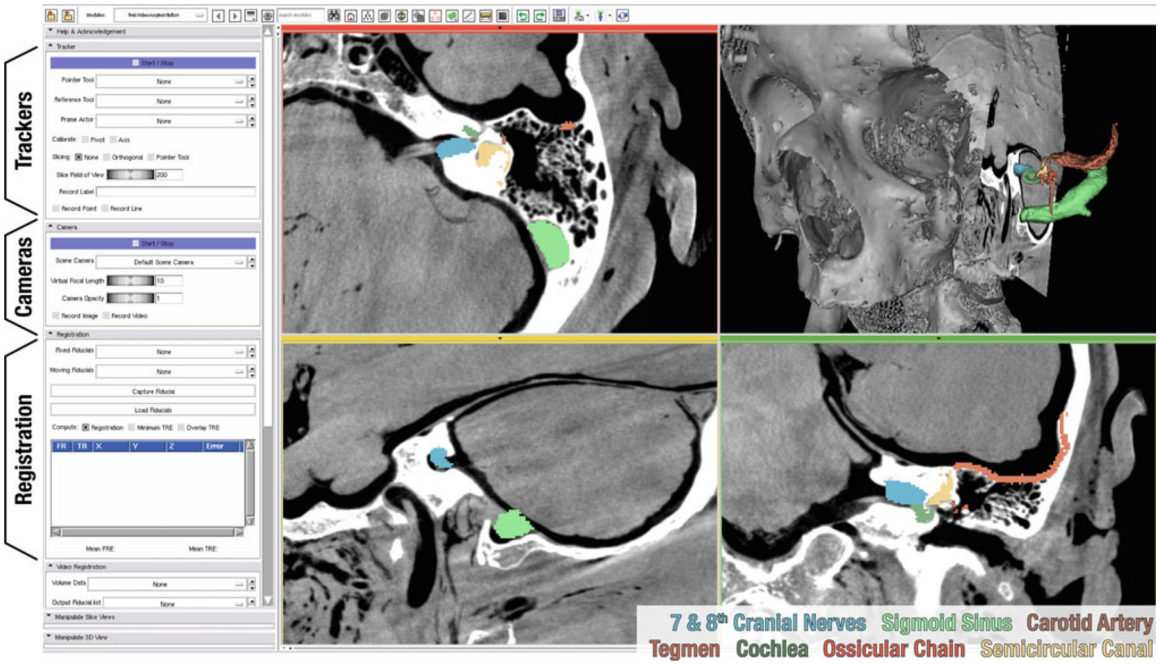


**Fig. 4.**

Scene showing CBCT guidance of temporal bone surgery. The *grayscale* images represent CBCT of a cadaveric specimen, and the *colored* structures were defined in preoperative CT, registered to the intraoperative scene. Module controls (shown on the *left*) provide toggling of functionalities and manipulation of parameters for tracking, video imaging, and registration. The controls are excluded (*cropped*) in subsequent figures

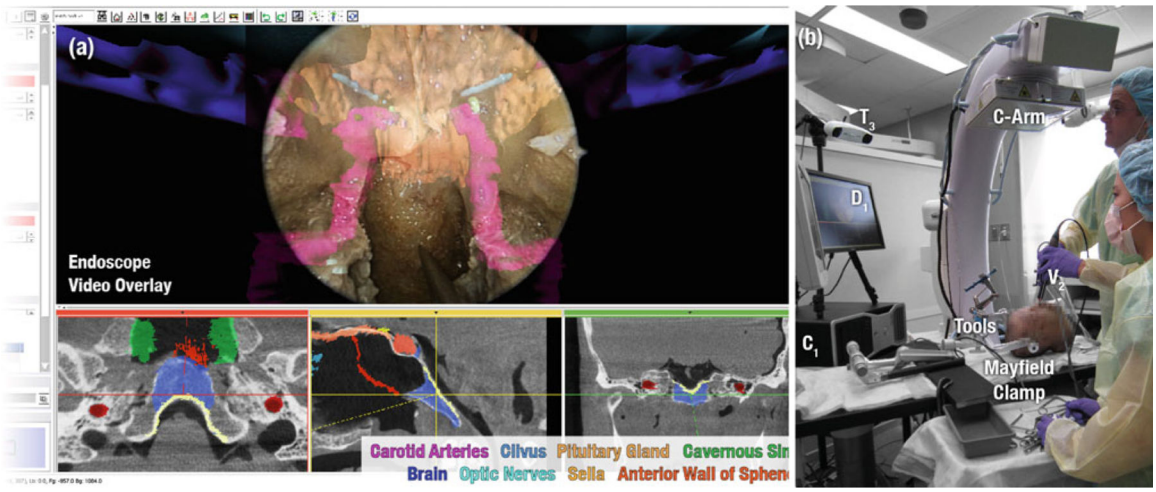


**Fig. 5.**

The virtual scene (**a**) and the experimental setup (**b**) of skull base surgery in cadaver

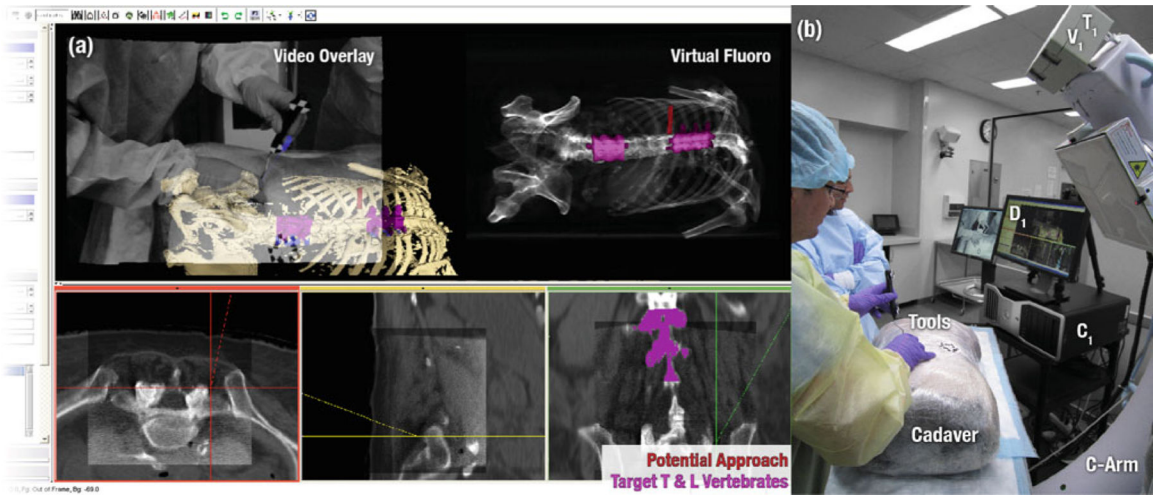


**Fig. 6.**

The virtual scene (**a**) and the experimental setup (**b**) of transpedicular spine surgery in cadaver