CustusX: A Research Application for Image-Guided Therapy

Release 1.00

Christian Askeland[1,2], Frank Lindseth[1,2,3], Ole Vegard Solberg[1,2], Janne Beate Lervik Bakeng[1,2], Geir Arne Tangen[1,2,3]

May 31, 2011

[1]SINTEF Technology and Society, Dept. Medical Technology, Trondheim, Norway, [2]National Center for 3D Ultrasound in Surgery, Trondheim, Norway, [3]Norwegian University of Science and Technology (NTNU), Trondheim, Norway

Abstract

CustusX version 3 is an in-house application for research in image-guided therapy with a focus on intraoperative use and ultrasound imaging. It is built upon open source toolkits such as the Insight Toolkit (ITK), the Visualization Toolkit (VTK), Qt and others. CustusX has a layered architecture based on services, and is extensible through the use of plugins. Key features are a user interface configurable by the user, an abstraction of the VTK pipeline, cross-platform video acquisition including temporal calibration, and a patient model. This paper gives an overview of the architecture and a description of the key concepts.

Contents

1 Introduction **2**

2 Architecture **2**

2.1 GUI organization 3
2.2 Services 4
Visualization Service 4
Patient Model Service 4
Tracking Service 5
Video Service 5
2.3 Plugins 6
Tubular Structures Registration 6
US Probe Calibration 6
Temporal Calibration 6

3 Conclusion **6**

1 Introduction

Several systems for image-guided therapy (IGT) exist today, both commercial [1, 2] and for research [3, 4]. CustusX version 3 is a research application focused on intraoperative ultrasound-based navigation. CustusX has existed in various shapes for ten years [5], but was rewritten from the ground in 2010. It is built on open source components, notably Qt [6], VTK [7], ITK [8], IGSTK [9], Eigen [10] and OpenIGTLink [11]. Some of the core visualization is shared with a commercial partner [1] through a closed source library.

CustusX is an in-house application for testing new functionality and supporting research studies in the operating room (OR). The users are both scientists and clinicians, and an important use of the system is to facilitate cooperation between the two groups and thus develop technology that is useful to the clinicians in their work. CustusX is designed to be extensible and customizable to the different clinical areas our group work with, in particular neurosurgery, laparascopy, bronchoscopy and endovascular therapy.

CustusX is designed to be a system for IGT, not a general toolkit for medical image analysis. Pre- and postoperative tasks such as segmentation are currently performed by applications such as OsiriX [12] and ITK-SNAP [13], thus interoperability with external applications is important.

This paper gives an overview of the CustusX architecture, and some key concepts that give strength to the system are described in more detail.

2 Architecture

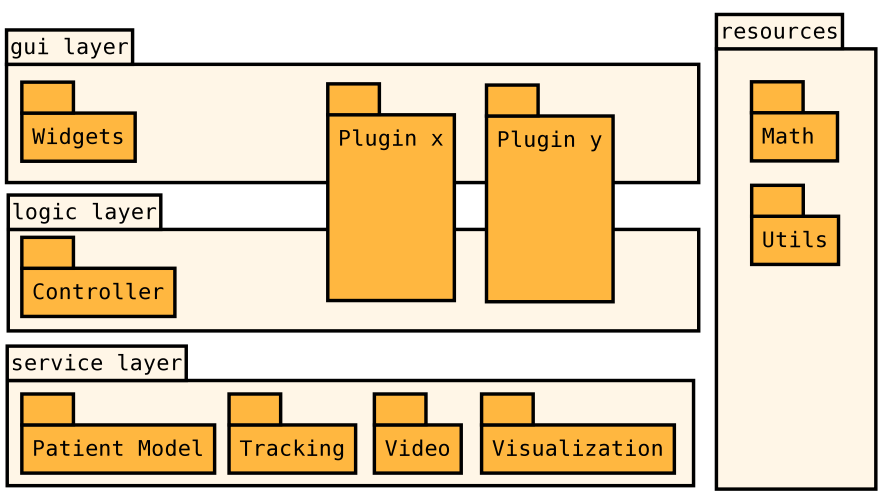


Figure 1: Overview of the CustusX architecture. The core is divided into three layers: The graphical user interface on top, the Logic Layer for control and functionality, and the Service Layer that provides basic services to the upper layers. The Resources Layer represent building blocks used in the rest of the system. The plugins extend the core with additional functionality.

Figure 1 provides an overview of the CustusX architecture. The *core application* consists of everything except the plugins and provides a basic framework able to visualize multimodal image data, tracking devices and video sources in 2D and 3D views. It can also visualize combinations of several preoperative data and an US probe with its real time video attached in the same 3D view.

The following *layers* are defined, each one dependent on the lower ones:

GUI Layer: Graphical user interface (GUI) consisting of widgets, extensible by plugins.

Logic Layer: Adds blocks of functionality on top of the services. A few core classes provide initialization of the services, everything else is added as plugins.

Service Layer: A collection of singletons [14] providing basic services: Patient Model, Tracking, Video and Visualization.

Resource Layer: A collection of common utility classes available to everyone, including a math library based on the Eigen linear algebra library [10].

The *plugins* extends the core by adding specific functionality. Examples of plugins are: Segmentation, Registration, Data Acquisition, 3D US Reconstruction and Calibration. They can be added by people not necessarily part of the core team, and can depend on other plugins. A plugin typically consist of a logic part that uses the services to perform an operation, and possibly GUI widgets for user interaction.

CustusX uses *Qt signals and slots* [6] in all layers, not only the user interface. This greatly simplifies module interconnections, as the software patterns Dependency Inversion [15] and Observer [14] are directly available in the language.

2.1 GUI organization

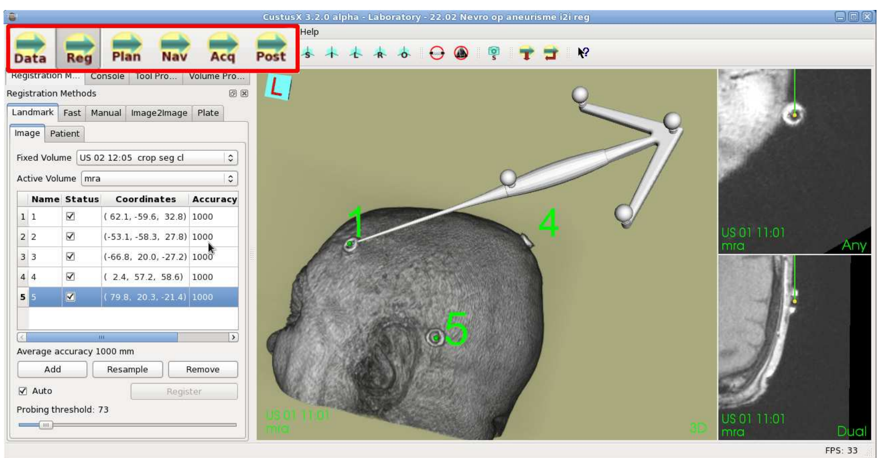


Figure 2: CustusX shown during the Registration Workflow Step. A `QMainWindow` organizes the desktop, with the Visualization Layout as the Central Widget to the right. The Registration Widget is added as a `QDockWidget` to the left, although it can be dragged around to any position. Note the enlarged Surgical Workflow Steps toolbar in the upper left corner.

The standard GUI is designed around a `QMainWindow` as shown in Figure 2. The user can configure both the dock widgets, toolbars and Visualization Layout (see chapter 2.2). The configuration can be stored this as a *Desktop* using built-in Qt features. The core system is not contsrained to using the `QMainWindow`, one can alternatively rebuild the GUI in any way using the widgets as building blocks.

A surgical procedure can be divided into several steps, called the *Surgical Workflow*. The following steps are defined and controllable by the user:

Preoperative Data	*Registration*	*Planning*	*Navigation*	*Intraoperative Acquisition*	*Postoperative Control*

The user can store a Desktop for each of these steps, and can select a step from a toolbar, giving access to all stored Desktops.

The concept of *Clinical Application* is defined as a state in the system, enabling customization for each clinical area. As an example, this is used to display Axial-Coronal-Sagittal (ACS) slices in either radiological (view from feet) or neurological (view from head) orientations.

2.2 Services

The services provides basic functionality that can be used by everyone in the system.

Visualization Service

Figure 3: The visualization pipeline. Data objects are visualized by `Rep` objects in `View` widgets. The Views are organized in groups and laid out on the screen in a rectangular grid using a `Layout`.

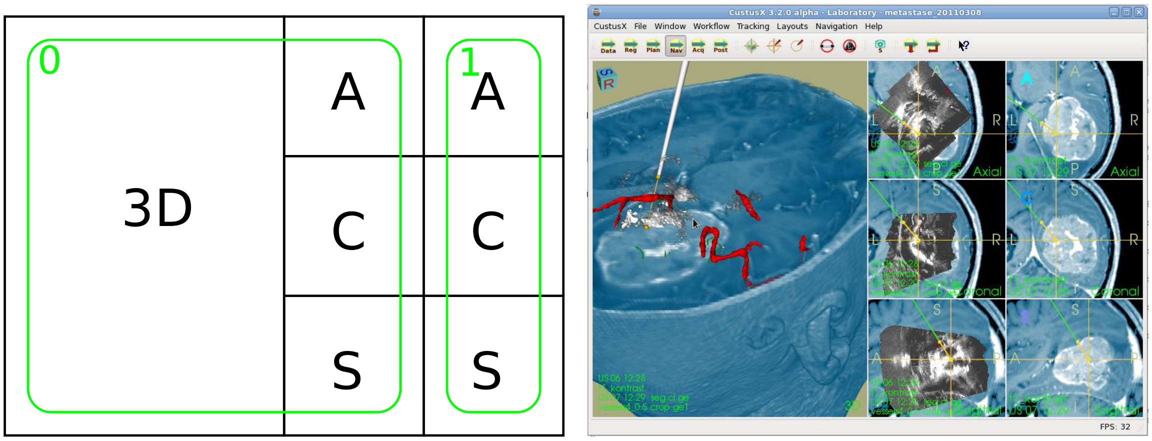
The Visualization Service is an abstraction built on top of VTK that aims to hide the complexity of that library. The pipeline is shown in Figure 3. The `Rep` (representation) class encapsulates visualizations of Data Objects, which is Service Layer entities such as volumes and tracking devices. A `Rep` contains the VTK pipeline from the data object to the `vtkProp`. Each `Rep` subclass implements a specific kind of visualization, for example:

VolumetricRep: Render a volume in 3D.
GeometricRep: Render a mesh in 2D or 3D.
ToolRep: Display a Tracked device, for example an US Probe, in 2D or 3D.
SliceRep: Show a 2D slice of a volume using some slicing specification (orthogonal or oblique). A tracking device is an input here, determining the interactive location of a slice.

The `View` class inherits from `QVTKWidget`, and thus serves as the link between Qt and VTK. A single `View` represents either a 3D scene or a 2D slice in that scene. Views that display the same data in different ways, for example 3D+ACS, are grouped in the class `ViewGroup`. Several ViewGroups together form a `Layout`, which organize the Views on screen using a `QGridLayout`. A layout editor allows the user to configure the `Layout`. A configuration example is shown in Figure 4.

Patient Model Service

The Patient Model Service contains information related to the current patient. This includes entities of data, such as volumetric data, mesh data, landmarks, labels, and temporal data such as video streams. Spatial, hierarchical and temporal relations between these entities are also available.



(a) Schematic (b) Screenshot

Figure 4: Example Layout configured with 3D+ACS View Group 0 displaying a MR volume with an US overlay and segmented vessels, and View Group 1 displaying only the MR volume. 4(a) shows a schematic of the layout, while 4(b) shows a screenshot.

All relations are stored and made available as historical information. This makes it possible to extract the spatial relation prior to and after a registration. The information can be displayed in the GUI as a registration history that can be used to validate registration algorithms.

Tracking Service

The Tracking Service is an interface to the physical *tracking devices*, such as navigation pointers, US probes and surgical instruments. This is built on top of IGSTK, which provides the actual connection to the various hardware, such as the Polaris and Aurora tracking systems from NDI [16].

Tracking devices are defined in an XML format, enabling addition of new objects without recompilation. The tracking devices operate within the space of the measurement volume of the tracking system, while the patient data exist in the patient space [17]. In order to navigate in the Patient Model, these two spaces must be registered to each other, and the relation stored in the Tracking Service. The actual registrations methods are provided via plugins.

Video Service

The Video Service is an interface to realtime image stream sources such as endoscopic, ultrasound and fluoroscopy video, along with means to connect to them.

The video acquisition might be available only on certain hardware and operating systems. Because of this, it is often desirable to run CustusX on another machine. The service uses the OpenIGTLink protocol to facilitate this communication. A standalone OpenIGTLink server built upon the Mac QuickTime Framework is currently used for videograbbing.

2.3 Plugins

Plugins provide various additional functionality not present in the core. Three plugins that we believe are of particular value will be described in this section.

Tubular Structures Registration

In addition to basic registration methods such as landmark registration, a tubular structures registration [18] is implemented in CustusX. In order to make this technique work in the OR, a dedicated user interface is created that incorporates all steps necessary, in particular segmentation, centerline generation and the registration algorithm itself. The plugin is being tested on brain and liver vessels, in addition to the bronchial structure versus a traversed bronchoscope path.

US Probe Calibration

A US Probe needs to be defined with some essential data before use. This includes calibration of the video image position [19, 20], the geometric shape of the US sector, depth settings and supported modalities such as B-Mode and Color Doppler. The calibration and definition are performed in MatLab, and then saved as XML readable by CustusX. This makes is it possible to add or reconfigure probes without recompilation.

Temporal Calibration

In order to perform US 3D reconstruction, the timestamps of the probe tracking data and the video frames must be synchronized. This temporal calibration is performed in two steps:

If the tracking and video grabbing run on different computers, the Network Time Protocol (NTP) [21] is used to synchronize the two computer clocks. When an Internet connection is available, the reference is a globally available atomic clock. If no connection is available, as is often the case in the OR, one of the computers is set up as a reference clock and communication is direct over an ethernet cable. The direct connection is also more robust, giving millisecond relative precision [21], and enables us to proceed as if tracking and video grabbing run on the same computer.

There is a shift arising from the fact that, for both tracking data and video frames, there is an unknown delay from the actual sampling to the timestamping. Temporal calibration difference between the sum of the shifts for those two streams [22, 23]. Preliminary results show that this shift is constant within at least 20ms, and it is found by performing a probe movement and then correlating the resulting tracking data with the structure movement in the video frames. This methods requires that enough structures are similar from frame to frame.

3 Conclusion

CustusX is an in-house research tool that aids in the development of new intraoperative techniques. It has already been used by our group to test out new registration methods and a new pituitary gland ultrasound probe. The clean and clear architecture helps the developers during development, while the simple and customizable gui helps the scientists in their interaction with the clinicians.

The XML-based customization of tracking devices and ultrasound probes has enabled fast integration of new hardware, while the OpenIGTLink video connection in combination with the temporal calibration frees CustusX from running on a particular machine while using arbitrary video sources.

We believe that the designs presented here can be useful in other projects, and that the CustusX project can contribute to the open source libraries on which it builds.

References

[1] Gronningsaeter A, Kleven A, Ommedal S, Aarseth TE, Lie T, Lindseth F, Lango T, Unsgard G. *SonoWand, an ultrasound-based neuronavigation system*. Neurosurgery 2000;47:1373-1379; discussion 1379-1380. http://www.sonowand.com/ 1

[2] Gumprecht HK, Widenka DC, Lumenta CB. *BrainLab VectorVision Neuronavigation System: Technology and Clinical Experiences in 131 Cases*. Neurosurgery 1999;44:97-104. 1

[3] Pieper S, Lorensen B, Schroeder W, Kikinis R *The NA-MIC Kit: ITK, VTK, Pipelines, Grids and 3D Slicer as an Open Platform for the Medical Image Computing Community*. Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging: From Nano to Macro 2006; 1:698-701. http://www.slicer.org/ 1

[4] Mercier L, Del Maestro RF, Petrecca K, Kochanowska A, Drouin S, Yan CX, Janke AL, Chen SJ, Collins DL. *New prototype neuronavigation system based on preoperative imaging and intraoperative freehand ultrasound: system description and validation* . Int J Comput Assist Radiol Surg. 2010 Oct 1 1

[5] Langø T, Tangen G, Mårvik R, et al. *Navigation in laparoscopy – Prototype research platform for improved image-guided surgery*. Minim Invasive Ther Allied Technol (MITAT). 2008;17:17–33. 1

[6] Nokia Inc. *Qt - Cross-platform application and UI framework*. http://qt.nokia.com/ 1, 2

[7] Schroeder, W.J., Martin, K.M., Avila, L.S., Law, C.C. *The Visualization Toolkit User's Guide*. Kitware, Inc. (2003) http://www.vtk.org/ 1

[8] Ibanez, L., Schroeder, W., Ng, L., Cates, J. *The ITK Software Guide*. Insight Software Consortium (2003) http://www.itk.org/ 1

[9] K. Cleary, L. Ibáñez, D. Gobbi, P. Cheng, K. Gary, S. Aylward, J. Jomier, A. Enquobahrie, H. Zhang, H.-s. Kim and M. B. Blake *IGSTK: The Book, 2.0 ed*. Signature Book Printing, Gaithersburg, Maryland, USA, 2007 http://www.igstk.org/ 1

[10] *Eigen C++ template library for linear algebra*. http://eigen.tuxfamily.org/ 1, 2

[11] Tokuda J, Fischer GS, Papademetris X, Yaniv Z, Ibáñez L, Cheng P, Liu H, Blevins J, Arata J, Golby AJ, Kapur T, Pieper S, Burdette EC, Fichtinger G, Tempany CM, Hata N. *OpenIGTLink: an open network protocol for image-guided therapy environment*. Int J Med Robot. 2009 Dec;5(4):423-34. http://www.na-mic.org/Wiki/index.php/OpenIGTLink 1

[12] Antoine Rosset, Luca Spadola and Osman Ratib *OsiriX: An Open-Source Software for Navigating in Multidimensional DICOM Images* . Journal of Digital Imaging Volume 17, Number 3, 205-216 http://www.osirix-viewer.com/ 1

[13] Paul A. Yushkevich, Joseph Piven, Heather Cody Hazlett, Rachel Gimpel Smith, Sean Ho, James C. Gee, and Guido Gerig. *User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability* . Neuroimage 2006 Jul 1;31(3):1116-28. http://www.itksnap.org/ 1

[14] Gamma et al. *Design Patterns: Elements of Reusable Object-Oriented Software* . Addison-Wesley 1994 2

[15] Martin RC. *The Dependency Inversion Principle* . C++ Report, May 1996 2

[16] Northern Digital Inc. http://www.ndigital.com/ 2.2

[17] Digital Imaging and Communications in Medicine (DICOM) PS 3 3.3-2009 2.2

[18] Reinertsen I, Lindseth F, Unsgård G, Collins DL, *Clinical validation of vessel based registration for correction of brain-shift* . Medical Image Analysis 2007; 11(6):673-684 2.3

[19] Mercier L, Langø T, Lindseth F, Collins DL *A review of calibration techniques for freehand 3-D ultrasound systems*. Ultrasound Med Biol 2005;31:449-471 2.3

[20] Hsu P-W, Prager RW, Gee AH, Treece GM *Freehand 3D Ultrasound Calibration: A Review*. In Advanced Imaging in Biology and Medicine. Springer Berlin Heidelberg 2008;47- 84 2.3

[21] D. Mills *Internet time synchronization: The Network Time Protocol* . IEEE Trans. Communications COM-39, 10 (October 1991), 1482-1493. http://www.ntp.org/ 2.3

[22] Treece GM, Gee AH, Prager RW, Cash CJC, Berman LH *High-definition freehand 3-D ultrasound*. Ultrasound Med Biol 2003;29:529-546. 2.3

[23] Rousseau F *A Novel Temporal Calibration Method for 3D Ultrasound*. IEEE Trans Med Imaging 2006;25:1108 2.3