

# Software Components for Medical Image Visualization and Surgical Planning

Y.P. Starreveld<sup>1,2,3</sup>, D.G. Gobbi<sup>1,3</sup>, K. Finnis<sup>1,3</sup>, T.M. Peters<sup>1,3</sup>

<sup>1</sup>Imaging Research Laboratories, The John P. Robarts Research Institute, London ON, N6A 5K8, Canada <sup>2</sup>London Health Sciences Centre, London, ON, Canada <sup>3</sup>Department of Medical Biophysics, University of Western Ontario, London, Canada

## Abstract

**Purpose:** The development of new applications in medical image visualization and surgical planning requires the completion of many common tasks such as image reading and re-sampling, segmentation, volume rendering, and surface display. Intra-operative use requires an interface to a tracking system and image registration, and the application requires basic, easy to understand user interface components. Rapid changes in computer and end-application hardware, as well as in operating systems and network environments make it desirable to have a hardware and operating system as an independent collection of reusable software components that can be assembled rapidly to prototype new applications.

**Methods:** Using the OpenGL based Visualization Toolkit as a base, we have developed a set of components that implement the above mentioned tasks. The components are written in both C++ and Python, but all are accessible from Python, a byte compiled “scripting language.” The components have been used on the Red Hat Linux, Silicon Graphics Irix, Microsoft Windows, and Apple OS X platforms. Rigorous object-oriented software design methods have been applied to ensure hardware independence and a standard application programming interface (API). There are components to acquire, display, and register images from MRI, MRA, CT, Computed Rotational Angiography (CRA), Digital Subtraction Angiography (DSA), 2D and 3D ultrasound, video and physiological recordings. Interfaces to various tracking systems for intra-operative use have also been implemented.

**Results:** The described components have been implemented and tested. To date they have been used to create image manipulation and viewing tools, a deep brain functional atlas, a 3D ultrasound acquisition and display platform, a prototype minimally invasive robotic coronary artery bypass graft planning system, a tracked neuro-endoscope guidance system and a frame-based stereotaxy neurosurgery planning tool. The frame-based stereotaxy module has been licensed and certified for use in a commercial image guidance system.

**Conclusions:** It is feasible to encapsulate image manipulation and surgical guidance tasks in individual, reusable software modules. These modules allow for faster development of new applications. The strict application of object oriented software design methods allows individual components of such a system to make the transition from the research environment to a commercial one.

## 1 Introduction

Applications of computer aided imaging technology to medical image visualization and surgical planning are continually increasing. These applications enable rapid image understanding and the planning of minimally invasive procedures that decrease operative time and/or morbidity.<sup>1,2</sup> In a realm of increasing possibilities, a number of guiding principles have become evident. Each application has common functions that must be performed: reading different image file formats, basic display and interaction functions, registration of images from different acquisition modalities, user interface, and interfacing with tracking systems are some of these functions. Once these basic functions are in place, the software that differentiates a minimally invasive robotic cardiac surgery planning application from a deep brain physiological atlas may be isolated. This isolation allows a focus on the unique aspects of a project to rapidly meet the imaging and planning demands made by new instruments and techniques such as dexterous micromanipulators and neuro-endoscopy.

### 1.1 Limitations of current systems

Currently, there are many medical image display and surgical planning applications – why is a new one desirable? The primary reason is these applications are either surgical support tools or image analysis tools. This dichotomy is compounded by the fact that the software architectures of each tend to be closed and idiosyncratic, making it impossible to combine the best components of separate applications.

Some applications are aimed primarily at the analysis of medical data (ANALYZE,<sup>3</sup> MNI,<sup>4</sup> MATLAB<sup>5</sup>). These applications have sophisticated image analysis tools, but are not realistically extensible to intra-operative use. Their monolithic architectures make it a daunting task to add new functionality, and make debugging difficult even when source code is available.

There are commercially available applications from a variety of vendors for intra-operative guidance. The image analysis tools provided with these applications are limited, and due to their proprietary nature, it is difficult to extend them with new capabilities.

An open-source application has recently been made available from the MIT AI Laboratory (The 3D Slicer<sup>6</sup>) that addresses some of these shortcomings. Unfortunately, it is bound to only a few operating systems (Solaris and Windows) and is primarily written in C++. It requires a logging system to enable reconstitution of the system state after crashes.<sup>7</sup> The scripting language used to bind the C++ classes to the user interface is Tcl, which has no object orientation and is limited in power as a language in its own right. As a result, it is difficult for a neophyte to extend the application. The 3D Slicer is also tied into a new file format, the MRML (Medical Reality Modeling Language), which is not widely used.

File format issues have also dogged other systems. While DICOM is emerging as a standard image format and communications protocol, it deals poorly with ultrasound images, time series of images, and is not extensible to encompass information storage requirements imposed by modern computer paradigms.<sup>8</sup> Many vendors support only portions of the standard, and there continue to exist a large number of legacy machines that have even lesser degrees of support. A system with any hope of clinical integration must be able to read and write DICOM files directly, but also to have a method to deal with the mentioned shortcomings.

Another problem is that the standardization of user interfaces has received little attention. Different commercial systems use different interfaces to perform exactly the same tasks, making it difficult to transfer knowledge of one system to another. Due to the constraints discussed above and to the difficulties encountered in attempting to re-engineer them, the demands of the user interface imposed by one task are necessarily present when the system is used in a different application. This has led to counterintuitive situations in which a system designed for intracranial navigation has been put into use for a spinal procedure.

In summary, although these individual tasks may be approached using various existing methods, it is not possible to combine the DICOM interactivity and instrument tracking abilities of a commercial surgical navigation system with the strong image analysis capabilities of MATLAB, and a custom written support module for a new tracked micromanipulator into a functional whole. Rapid and unpredictable advancements in computer hardware, computer graphics, medical imaging and surgical instrumentation are therefore difficult to harness with the current crop of monolithic systems.

## 1.2 Improvements made by the present system

By addressing each one of these areas in turn, we have arrived at a platform for image viewing, surgery planning and intraoperative guidance and support that is flexible, extensible, and hardware independent. Its user interface components have been designed with specific applications in mind, providing clinicians with the ability to access information in a fashion that is intuitive and context driven, while keeping common general principles in place for tasks that are common to the individual applications.

The system has been designed to allow applications to be quickly prototyped for surgical planning and intraoperative data acquisition and guidance. Due to its modular nature, user interface development and modeling of new surgical tools are easily accomplished.

Many of the components of the system and the control of its information flow are written entirely in Python. This object oriented byte compiled language allows those with limited computer experience to construct functioning applications rapidly, by dramatically shortening the compile debug cycle. Components that are initially written and debugged in Python may then be translated to C++ when performance requirements dictate.

Thanks to these general design principles, the components have enabled the rapid design and deployment of applications that exploit the following four areas:

**Multi-modal Image Fusion** — Imaging modalities have complementary strengths. Tissue contrast of MR is better than that of ultrasound, but MR is not as fast or as portable. Three dimensional computed rotational angiography provides exquisite resolution of vascular structures, but MR provides better soft tissue contrast. By compartmentalizing the modality specific details of the acquisition in a software component, it becomes possible to use the power of the entire application on an arbitrary modality or combination of modalities.

**Physiologic Data Fusion** — Microcellular recordings from neurons in the deep brain are not images themselves, but benefit from being visualized in an environment of CT or MR images.

**Real Time Video Overlay** — Endoscopes provide direct visualization of surrounding structures. Providing extra context from other imaging modalities for the positioning of endoscopes prior to their insertion and orientation during times of reduced vision combines both the tracking and image display facilities of the platform.

**Bridging Commercial and Research Environments** — The component nature of the system allows compartmentalization of the code. This allows each component to be tested in isolation and provides a vehicle for the certification and licensing of portions of an application.

This paper reviews the specific design motivations in each area and the manner in which they were implemented. A description of applications to some different planning and surgical procedures follow.

## 2 System Architecture

### 2.1 Philosophical Issues

We approached the necessary design decisions of our planning platform at a number of levels: computer hardware, low level graphics libraries, visualization libraries, user interface components, and the language used to structure and design the end applications. A benefit of this division into different levels of abstraction has been the isolation of different categories of dependencies. For example, much of our work has taken place at the relatively high level of creating additions to the visualization libraries and manipulating user interface components. At a lower level, portions of the code of the tracking system interface had to be written directly to a particular operating system and a particular hardware tracking implementation. The end result is that if one is adding support for a new tracked neuro-endoscope, it requires work only at the most abstract level, despite the tracker being plugged into a Linux workstation for development, but into a Macintosh laptop in the operating room. Conversely, should a new type of tracking system be developed, the only code that must be written is that which handles the hardware differences between this tracking system and one which was previously supported. The neuroendoscopy application itself need not be modified.

### 2.2 Hardware Platform Independence

From previous experience with surgical planning and visualization software, it is clear that hardware platforms come and go as the march of increasing capability continues. Our aim has been to be as flexible as possible in our choice of target platform. As far as hardware and operating system platform compatibility are concerned, we have five categories of components: external hardware interface (tracking systems, video input), image processing, image display, user interaction and process control.

**External hardware** requires device specific software. A tracking system may use the serial port on both Windows and Linux, but accessing the serial port hardware from these different operating systems requires operating system specific code.

**Image processing** lends itself well to abstraction. Provided that the data can go to and from the algorithm reliably, the implementation of an image-processing algorithm rarely demands anything more than a C or C++ compiler. Some issues arise with respect to the multi threading of algorithms and taking advantage of machine specific features, but these are dealt with below.

**Image display** has traditionally been quite device specific. High-end graphics workstations have provided ultimate standards of performance for some time. On the other hand, generic personal computer architectures with highly capable, low cost graphic boards have become viable, cheaper alternatives. Bridging the gap between the two has become possible due to the ascension of OpenGL, a standard interface to graphics rendering hardware that is now available on everything from the largest multiple-CPU graphics workstations to the least expensive PC graphics boards.

**User interface** components, such as buttons, sliders, windows and mouse interaction are some of the largest barriers to complete platform independence. Tk<sup>9</sup> and Java<sup>10</sup> come the closest with their “widgets” and the Java Foundation Classes/Swing library. We have taken an agnostic approach, writing to the lowest common denominator of these different API’s, recognizing that different end users will have unique requirements, but providing a ground level of functionality throughout.

**Process control** components are those which define the character of the entire application. They determine which lower level components are used and how they are integrated. We have chosen to implement them entirely in Python. This byte

compiled, object oriented language is powerful, yet accessible to the neophyte. It is available for almost any combination of hardware and operating system, allowing this level of component to be completely devoid of device dependencies.

### 2.3 External Hardware Interfaces

We have written components that acquire information directly from external sources such as tracking devices for operative instruments and video from ultrasound machines. The trackers and video sources are considered separately due to the difference in bandwidth each requires.

The tracking systems that we use are the POLARIS (Northern Digital, Mississauga, Ontario) which uses infrared cameras to track light emitting diodes, and the Flock of Birds (Ascension Technology Corporation, Burlington, VT) which uses a magnetic field. Both interface to the host computer using a serial port, and both provide essentially the same information albeit using different methods and vastly different protocols. Since the end user is concerned only about the ability to get a transformation for a tracked instrument, and what the precision of that transformation is, the tracking modules were written to have identical APIs. Thus, the user can create an application using one system, and then switch to using another without having to change any source code. This transparency is obtained by making each type of tracker a subclass of a more general tracker object.

At the machine level, flexibility is attained using conditional compilation within the individual functions that interact with the serial port. This occurs as part of the configuration and installation process. In this way, appropriate serial port calls are used whether the code is being compiled on a version of Unix, Windows, or the Macintosh.

Video acquisition requires considerably more bandwidth than can be supplied by a serial port. Therefore, a special purpose interface connected directly to the PCI bus, or to an IEEE-1394 (Firewire) connector is required. Currently, we only have a component that gets input from a Video for Windows compliant source. Once again, the API has been designed so that future changes to allow input from other sources will not break the functionality of applications using the current video input component.

### 2.4 Graphics Library

Image processing and image display functions form the core of medical image display and surgical planning applications. There are a number of different packages available which will perform these functions, such as AVS (Advanced Visual Systems, Waltham, MA) or ANALYZE,<sup>11</sup> but their cost, lack of source code availability or monolithic architectures made them unsuitable. The Visualization Toolkit (vtk)<sup>12</sup>, a class library written in C++, most closely matches our requirements. It consists of two pipelines; one for image processing, and one for graphics. Its architecture embodies a strict object oriented design methodology,<sup>13</sup> which allows an easy extension of the system's functionality.

The image processing pipeline in vtk is designed to stream data through a series of filters, the configuration and parameters of which may be varied. It has facilities to split large chunks of data into pieces to handle memory constraints, or so that they can be sent to multiple processors. The image pipeline will compile and run on just about any machine with a C++ compiler and POSIX compliant threads, making it possible to implement very high performance applications on multi-processor hardware.

The processing information for display in vtk may be visualized as a process of linking a data source to one or more processing filters and displaying the output. A very simple example would involve the reading of a volume dataset, the extraction of one slice of that dataset, and the display of that slice on an appropriately oriented plane in a three-dimensional display. Once this processing architecture is specified, one need only change the attributes of the slice extraction filter and request a pipeline update to have all the appropriate computations along the way redone and the new slice displayed on the screen.

**Performance** can be a concern, especially when an object oriented software library is weighted heavily on the side of logical design and multiply inherited class structures. The hardware can end up spending a lot of time calling functions and allocating memory for objects rather than processing data. Vtk was initially designed as a demonstration of object oriented design technique, and as such, performance was not a primary goal. The component-based architecture is an aid to optimization, however, since it is easier to isolate the code upon which effort should be expended. The multiple thread architecture which is a part of the processing pipeline in vtk is another feature which can provide dramatic improvements in performance in return for relatively little additional programming effort.

The transition from image processing to image display requires consideration of the geometry that governs that display. When using vtk, that transition is the transition from the imaging pipeline to the graphics pipeline. The graphics pipeline is

written to use OpenGL, from Silicon Graphics. Implementations of OpenGL exist on many platforms, with many of those platforms offering significant hardware acceleration of most functions.

The output of the graphics pipeline is the windowing system of the target computer platform. Operating system-specific classes are present in vtk for the X window system and for Microsoft Windows. We have added ones for Apple OS X. Encapsulating the system specific code in these subclasses, focuses the work of supporting a different windowing system here, allowing entire applications written using vtk to run unchanged across different hardware and operating systems.

## 2.5 User Interface

The user interface requires more than displaying graphics in a window. Methods for the user to interact with and modify the display of the information must also be provided. Natively, vtk provides components that manage the interaction of the user with the window using the keyboard and the mouse for both the X and Windows windowing system. We have added a similar component to manage this interaction for the Apple OS X operating system. This mechanism provides one with the raw tools to trap mouse clicks on individual objects in the display and convert them to display and three dimensional world coordinates. Using this mechanism, it is possible to write code to manipulate objects in this display and to alter more global settings, such as the location and orientation of the virtual camera from which the display projection is computed.

More complex interaction is also required. For example, the setting of a window and level when filtering a dataset through a lookup table for display, or being able to specify source or destination files in the computer's file system. These tasks have traditionally been the province of user interface toolkits typically provided with a computer system. These toolkits are often purposely made different from those offered by competitors for marketing reasons.

In the quest for ultimate platform flexibility, a number of cross-platform user interface toolkits have been made available, including Tk from Scriptics and the Java Swing library from Sun Microsystems. To this point, we have written all of our components so that they do not have dependencies on any one interface toolkit. This has been achieved through the use of an abstraction layer between the components to be manipulated and the events as generated by the user interface toolkit. This approach has worked very well for items which fit naturally into the "button, slider, entry field" paradigm.

More sophisticated user interface components have been engineered within vtk itself. This approach has the advantages of being completely cross platform, extremely flexible, and requires fewer context shifts of the user while interacting with the application.

### 2.5.1 Haptic feedback

Feedback by modulating force on the user's hand has been incorporated into the user interface of our image guided neurosurgery platform. When using an input device capable of force-feedback, this modality is used to heighten the experience of three-dimensional interaction. When moving objects about on the screen, often there is a constraint on the ways in which it makes sense for the object to move. The haptic device models these constraints for the user. Haptic feedback can also be provided during the maneuvering of modeled surgical tools. When the tool moves through the brain parenchyma, resistance can be provided based on any number of physical or functional tissue attributes. Currently supported haptic devices include the MouseCAT and the PenCAT/Pro, both from Haptic Technologies.<sup>14</sup>

### 2.5.2 Stereoscopic Projection

A very useful way of presenting renderings of three-dimensional scenes is with stereoscopic projection. In the planning stage, either liquid crystal glasses, or a liquid crystal polarizer on the monitor can provide separate images to the left and right eyes. This technique simplifies the viewing of three-dimensional displays on a monitor, especially when they have large numbers of objects in them.

There is a natural extension to the operating room, since the operating microscope is stereoscopic, and any image fusion with this device requires the generation of separate projections for each eye, allowing guidance for the microscopic portion of an operation.<sup>15</sup> Our system is designed from the ground up to support stereo visualization on any video board that provides it.

## 2.6 Process Control

An important design goal was that the system needed to be modifiable by those without sophisticated software engineering backgrounds. Frequently in a research setting, the length of time a student has to complete a project is such that spending two or three months getting familiar with the software in order to be able to get started doing something is too long. Requiring a

working knowledge of C++ excludes much of the target population also. Python<sup>16</sup> has been developing since 1990, and has become an extremely popular, cross platform scripting language. Its rivals are Perl and Tcl for simpler applications, but Python has proven its worth for systems consisting of thousands of lines of code which are maintained by multiple programmers. The prime reason for this is the object model that it implements which allows the compartmentalization of code into more easily maintainable pieces.

Another feature of Python is the ease with which one can extend the language. By writing modules of code in either Python or C++, it is possible to add unlimited functionality. A particularly nice aspect is that it is possible to write modules in C++ which are indistinguishable from Python modules by end users' applications (and vice versa). This makes it possible to rapidly develop and debug components in Python, and then translate portions of them to C++ if performance requirements so dictate. The majority of the code ends up being written in Python, however, since in creating similar applications, one half to one fifth as much source code and development time are required compared to C++ or even Java.<sup>17</sup>

A final benefit has been the ability to integrate the Python interpreter into existing commercial applications. Doing so creates a bridge whereby modules from a research environment can be incorporated into a commercial surgical planning system — a bridge that benefits both sides.

## 2.7 File Formats

In order to work with images on a computer, the first step is to read them. Many extant imaging consoles have no simple access to networking, and are restricted to a variety of physical media for image export. The rapid transition to networked, standards based consoles has changed the focus from the media to the file format. This is a problem that has received much attention with the adoption of a number of medical imaging standards in the form of DICOM-3 (Digital Imaging and Communications in Medicine). Although many formats are called DICOM-3, each vendor has implemented specific extensions to the standard, which form obstacles to the seamless interchange of images.<sup>18</sup> In its current form, DICOM-3 does not provide support for multi-dimensional images, addressing these only as groups of slices,<sup>19</sup> or storing a color reconstructed image as a "screen dump." There are active efforts underway to develop three dimensional image support as an extension to DICOM-3.<sup>20</sup> In the meantime, we have adopted the MINC (Medical Image NetCDF) file format,<sup>21</sup> which, in our usage is simply another name for the NetCDF file format (Network Common Data Form) distributed by the Unidata Program Center.<sup>22</sup> This format has a number of advantages not least of which is that it has been around for ten years. It is machine and media independent, being accessible from servers, workstations and PCs. It is self-documenting. Applications can query the datafile to determine the attributes of a datum, and handle it appropriately. New attributes can be added to old files, and old files can be gracefully handled by new applications. The format documents a history of each file automatically. Finally, the dimensionality of the dataset is arbitrary. Slices, volumes, and time series of volumes can be stored efficiently.

The majority of our images arrive from scanners in various proprietary and DICOM formats. Filters have been written which allow us to read these formats. Exportation of raw data or of DICOM files is also possible, but any additional attributes are written to a separate text file.

## 3 Results: Applications

Once the functional specification of the basic components had been drawn up, the creation of applications that drew on and extended these components could occur. One of the most common tasks is the recreation in a virtual environment of a surgical instrument, so that planning of surgical procedures can be done more realistically, and so that intraoperative support can be done more precisely.

### 3.1 Modeling of Surgical Instruments

Neuroendoscopy is the intracranial use of an endoscope to be able to perform minimally invasive cranial surgery under direct vision. It is desirable to have a neuroendoscope integrated with a surgical planning and support system in order to be able to plan an entry point that will allow access to all required structures intracranially. Provision of support required the fashioning of a holder to allow an actively tracked tracking block to be attached to the endoscope. The dimensions of the endoscope and its relationship to the holder were then used to create a replica of the scope that could be used as a virtual instrument in the

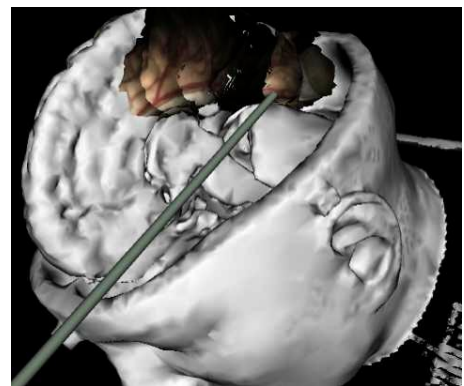


Figure 1: Tracked video from neuroendoscope overlaid on 3-D CT phantom image.

operating room. An image of the scope being tracked in relationship to a patient is shown. This setup has been used to plan one procedure.

The video import component has also been used in order to integrate the information coming from the camera on the endoscope with segmented CT scans of a phantom, potentially providing another source of orientation in times of reduced vision due to bleeding. An image of the video information overlaid on the phantom is shown in figure 1.

Another application along these lines involves the planning of minimally invasive coronary artery bypass surgery. This relatively new procedure allows the revascularization of the heart using the internal mammary artery without having to stop the heart and place the patient on cardiopulmonary bypass. Access to the surgical site is via ports placed in the chest wall, permitting the entry of two robotic manipulators and an endoscopic camera. The robotic manipulators have limitations in their ranges of movement such that their interaction with the patient's geometry is critical in determining the placement of the access ports in the chest wall. Improper placement of these ports will not allow completion of the procedure. In order to assist in the determination of their placement, a virtual minimally invasive cardiac surgery simulator was made<sup>23</sup>.

Components that were needed for this task were virtual copies of the robotic manipulators and of the endoscope. These were tracked using the Flock of Birds tracking module. An image comparing the virtual view to the physical view of a chest phantom is shown in figure 2.

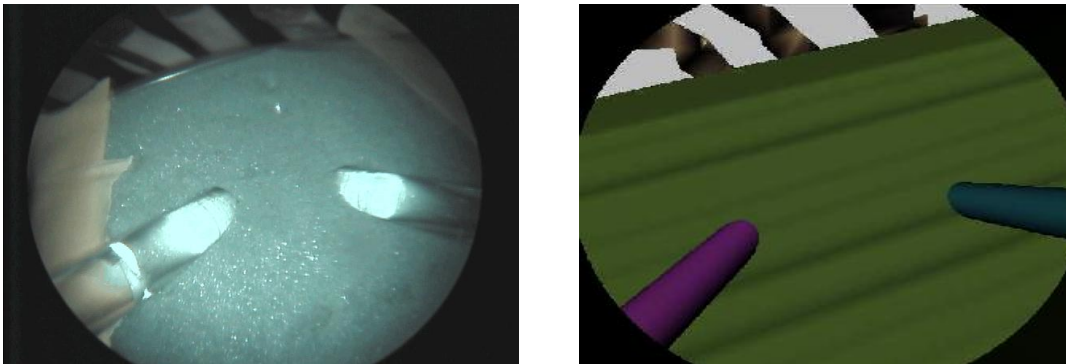


Figure 2: Comparison between physical and virtual view of heart phantom as implemented in a cardio-thoracic surgical simulator based on ASP.

### 3.2 Multimodality Image Fusion

The skull base represents one of the most anatomically complex areas of the human body. In a very small space, there is intricate bone structure laced with blood vessels and cranial nerves. Operating on pathology in this region requires the removal of bone and the sparing of soft tissues located within that bone. Access to the region is difficult, and space is constrained. Applications for neural navigation abound. Due to its facility with multi modality data, our system can register and display the anatomy as it exists pre-operatively, displaying tissues, bone and vessels with combined MRI, CT and computed rotational angiography, allowing the simulation of various approaches. Intra-operatively, registering functional data from a tracked nerve stimulator adds another dimension.

Registration of intra-operative ultrasound with the preoperative MRI demonstrates changes in the anatomy during the operation.<sup>24</sup> This has particular application in the removal of large skull base tumors, which are typically approached by "gutting" them from the inside, and then removing the capsule that is left. Without intraoperative guidance, it is impossible to know where one is with relation to other structures, necessitating a very slow, methodical approach, to avoid damage to cranial nerves in the region. Registration of the anatomy as it is changing provides many of the advantages of intraoperative MRI, but at a fraction of the cost.

Attaching a tracker to an ultrasound transducer, registering it to the patient and the patient's preoperative MRI and calibrating the probe, provided the rudiments of an integrated intraoperative ultrasound imaging facility. Adding the display of the video stream from the ultrasound machine to an appropriately oriented virtual ultrasound slice in the three dimensional display provided an integrated image. Currently, work is proceeding on allowing the preoperative information to be warped to match the intraoperatively acquired ultrasound, giving high quality, up to the minute imaging feedback at a fraction of the cost of an intraoperative MRI installation. Figure 3 shows the registered ultrasound and MRI images of a phantom.

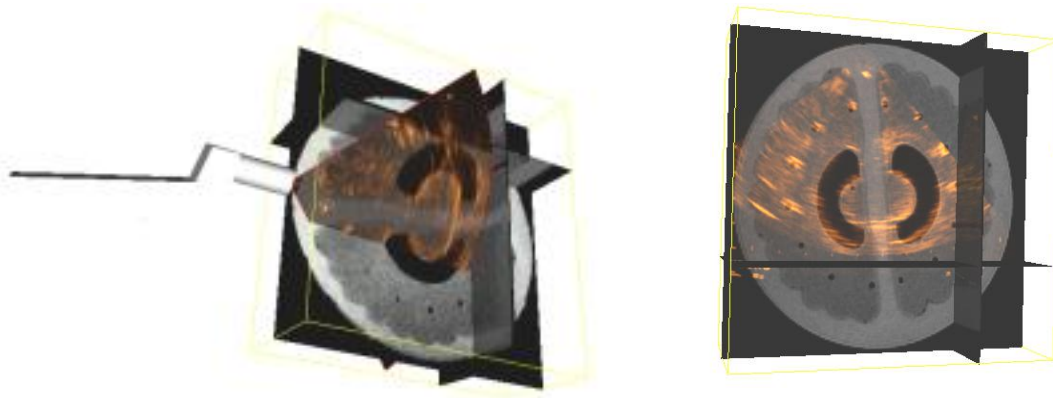


Figure 3: Ultrasound overlaid on MRI of stylized brain phantom. Left – US plane oriented at arbitrary orientation with respect to the MR volume; Right, US and MRI planes co-incident.

### 3.3 Physiologic Data Fusion

Neurosurgery is concerned with the precise localization of functional anatomy. Unlike other body tissues in which form is quite predictive of function, neurons, which are identical on MR or CT imaging, assume entirely different functions depending on their locations. Planning surgical intervention must take this into account, especially in the deep brain. Surgical intervention in this area for the treatment of Parkinson's disease and other movement disorders involves the precise placement of a radiofrequency lesion or a neurostimulator. The target site is selected based on landmarks seen on preoperative MRI or CT images, but can not be seen directly on the images, since its precise definition is functional rather than anatomic.

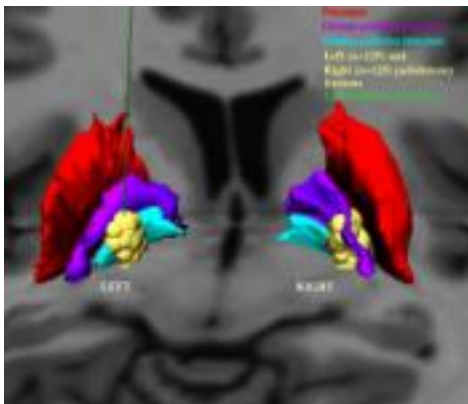


Figure 4: Physiologic atlas displayed in context with deep brain structures and MRI data.

There is no universally accepted methodology for lesion placement, but there is mounting evidence that physiologic rather than morphologic localization is most effective in guiding lesion placement.<sup>25</sup> This has led to our work that is concerned with the integration of electrophysiological atlas databases with individual patient MR volumes.<sup>26, 27</sup>

Further work is required to characterize the physiological variation in the basal ganglia.

This atlas was developed using the same basic set of software components as those used above. Specific developments for the physiologic atlas were modules to translate the individual data points into geometric objects, models of the recording electrode allowing its trajectory and position to be modeled for precise location of each data point, and a unique user interface component to facilitate searching the database using the homunculus representation. Figure 4 shows a screenshot of the atlas in operation.

### 3.4 Frame based stereotaxy

Any modern surgical planning and guidance software should include stereotactic capability. The history of stereotactic localization is long, going back to 1908 with the application of a frame to localize coordinates intracranially.<sup>28</sup> With the advent of CT and MRI neuroimaging, dataset registration by means of imaged fiducials has become popular. These fiducials may include a frame attached to the patient's skull with pins, bone mounted fiducials without an attached frame,<sup>29</sup> skin mounted markers,<sup>30</sup> or a repeatably remountable frame, which is completely non-invasive.<sup>31</sup> Registration is also possible using laser scanners,<sup>32</sup> which dispenses with the need to use markers of any type. Each method has applications for which it is best suited. The gold standard for registration accuracy requires a bone mounted solution,<sup>33</sup> either a frame or individually implanted markers. The major downside is their relative invasiveness.

There has been a recent trend toward developing "frameless" stereotactic systems that are designed to avoid the invasiveness of rigid frame fixation. However, even these systems require that the head be rigidly fixed at the time of craniotomy, both for surgical purposes and for the maintenance of image registration.

Image guided surgery is best viewed as a continuum with rigid frame based systems at one end and frameless systems at the other. In procedures such as thalamotomy, pallidotomy and stereotactic biopsy a probe must be held in fixed relationship to the head. A rigid frame based system allows a convenient method of obtaining accurate fiducial registration as well as a platform for holding and directing the necessary brain probes or biopsy tools. On the other hand, a surgical procedure to remove a large skull base tumor does not require the sub-millimetre accuracy of rigid fiducials, but does require relatively unencumbered access to the surgical site which could be compromised by the presence of a frame.

The issue comes down to a compromise between invasiveness and accuracy. Bone mounted or frame based fiducials yield an overall accuracy of about a millimetre, compared with 3-5 millimetres for noninvasive frames such as the Laitinen stereoadapter.<sup>34</sup>

Our system supports both frameless and frame based registration. It incorporates an automatic frame-based fiducial localization algorithm to allow real-time conversion of image coordinates to Leksell coordinates.

Frame based stereotaxy is the method of choice for deep brain functional procedures where high accuracy, patient immobilization and rigid instrument fixation are required. Its accuracy is dependent on precise positioning of the frame, minimization of spatial errors intrinsic to the imaging modality used, the mechanical properties of the frame and the planning computation. The planning computation depends on the accurate selection of fiducial points in the image. Errors in selecting these points propagate through the computation to degrade the final result and are the largest source of application error. Our algorithm automatically identifies the fiducial markers and computes a transformation based on the volumetric configuration of the fiducial marker set, rather than on individual slices through the markers. It has been validated with a phantom study comparing computationally derived target locations with locations found mechanically using a Leksell G frame and arc. The mean vector error is  $0.5 \pm 0.3\text{mm}$  for CT, and  $0.6 \pm 0.3\text{mm}$  for MR. The technique has particular application to MR where the algorithm's automatic monitoring of image acquisition quality helps to prevent loss of accuracy. It has been used on over 200 clinical MRI and CT datasets with a mean RMS error of less than 0.2 mm.

The frame finding algorithm has been used extensively by the deep brain physiologic atlas, where it has dramatically increased the accuracy and reduced the time required to code each one of the more than 5000 points in the database.

The frame finding module has been validated and licensed for inclusion in a commercial image guided surgery system, making use of the incorporation of the Python interpreter into the commercial application as mentioned earlier.

## 4 Conclusions

This paper describes our approach to the development of a robust platform for the development of image viewing and surgical planning applications that allows the easy integration of images from multiple sources, as well as the efficient development of new modules. Our platform is designed to handle the integration of image-oriented physiological data such as functional MR and PET, but also data from microelectrode recordings obtained during pallidotomy and thalamotomy procedures.

The platform is hardware independent and rapidly extensible. End applications have been developed to support stereotactic procedures; neurosurgical procedures such as biopsy, pallidotomy and thalamotomy with both anatomic and functional guidance; real-time ultrasound integration with pre-operative imagery; minimally invasive cardiac surgery planning and neuroendoscope integration. The system has support for a wide variety of input devices, including tracked pointers, head tracking, video and haptic feedback devices.

The user interface of each component is specifically tailored to it, and reuse of these components enforces a consistency of style from application to application. The end result is a set of applications that are intuitive to use, both for the experienced user, and the neophyte.

The development time for new applications has been dramatically reduced. For example, it was possible to extend the cardiac surgical planning application to use a 4-dimensional time series of images to allow the interactive exploration of an animated acquisition with the investment of less than an hour.

## Acknowledgments.

We wish to acknowledge the contributions of our colleagues in our group and in the Imaging Research Labs at the John P. Robarts Research Institute. In particular we wish to thank Dr. Piotr Slomka, Dr. Damini Dey, and Ms. Adeline Chiu for helpful discussions while creating the illustrated endoscopic and cardiac applications with the components, and Dr. Hua Qian for his contributions to the software base. This work was supported by the Canadian Institutes for Health Research grants, MT 11540 and GR 14973, as well as the Institute for Robotics and Intelligent Systems (IRIS) National Centre of Excellence.

## References

---

- <sup>1</sup> Paleologos, T.S., Wadley, J.P., Kitchen, N.D., Thomas, D.G.T., Clinical utility and cost-effectiveness of interactive image-guided craniotomy: clinical comparison between conventional and image-guided surgery. **Neurosurgery** 47(1):40-48, 2000
- <sup>2</sup> Biglioli P, Antona C, Alamanni F, Parolari A, Toscano T, Pompilio G, Polvani G, Minimally invasive direct coronary artery bypass grafting: midterm results and quality of life. **Annals of Thoracic Surgery** 70(2):456-460, 2000.
- <sup>3</sup> Mayo Clinic. *ANALYZE Software*. [http://www.mayo.edu/bir/analyze/ANALYZE\\_Main.html](http://www.mayo.edu/bir/analyze/ANALYZE_Main.html).
- <sup>4</sup> MNI. *MNI Software*. <http://www.bic.mni.mcgill.ca/software/>
- <sup>5</sup> The Mathworks, Inc., Natick, MA, <http://www.mathworks.com>
- <sup>6</sup> The 3D Slicer. <http://www.slicer.org>
- <sup>7</sup> Gering, D., A System for Surgical Planning and Guidance using Image Fusion and Interventional MR. MIT Master's Thesis, available online at <http://scooby.ai.mit.edu/pubs/index.html>, 1999
- <sup>8</sup> Chimiak W, Lopez D, An open architecture for medical imaging systems. *J. Digital Imaging*, 13(2):60-69, 2000.
- <sup>9</sup> The Scriptics website: <http://www.scriptics.com>
- <sup>10</sup> The Java website: <http://java.sun.com>
- <sup>11</sup> Robb, R.A. and Hanson, D.P.: The ANALYZE software system for visualization and analysis in surgery simulation. Book Chapter in: Computer Integrated Surgery. Eds. Stephane Lavalle, Russ Taylor, Greg Burdea, and Ralph Mosges. 1993
- <sup>12</sup> Schroeder W, Martin K, Lorensen B: *The Visualization Toolkit, 2<sup>nd</sup> Edition*. Upper Saddle River, NJ: Prentice Hall, 1997
- <sup>13</sup> Rumbaugh J, Blaha M, Premerlani W, Eddy F, Lorensen W: *Object-oriented modeling and design*. Saddle River, NJ: Prentice Hall, 1991
- <sup>14</sup> Haptic Technologies, Montréal, Québec, Canada, [www.haptech.com](http://www.haptech.com)
- <sup>15</sup> Edwards PJ, Hawkes DJ, Hill DL, Jewell D, Spink R, Strong A, Gleeson M: Augmentation of reality using an operating microscope for otolaryngology and neurosurgical guidance. *J Image Guid Surg* 1(3):172-178, 1995.
- <sup>16</sup> The Python website: <http://www.python.org>
- <sup>17</sup> Salus PH: *Handbook of Programming Languages Volume 3: Little Languages and Tools*. Macmillan Publishing, 1998.
- <sup>18</sup> Brown NJ, Britton KE, Plummer DL *Int J Med Inf* 1998 Feb;48(1-3):227-38, Standardisation in medical image management.
- <sup>19</sup> The 1998 DICOM draft standard, located online at <ftp://ftp.nema.org/medical/dicom/final>
- <sup>20</sup> The National Electrical Manufacturers Association home page: <http://www.nema.org/nema/medical/dicom/>
- <sup>21</sup> Neelin, P: *The MINC Programmer's Guide*. Online at <http://www.bic.mni.mcgill.ca/software/minc/>
- <sup>22</sup> University Corporation for Atmospheric Research Unidata Program Center, Boulder, Colorado, USA
- <sup>23</sup> Chiu AM, Dey D, Drangova M, Boyd WD, Peters TM: 3-D Image Guidance for Minimally Invasive Robotic Coronary Artery Bypass" Heart Surgery Forum, #2000-9732, 2000 <http://www.hsforum.com/vol3/issue3/2000-9732.html>
- <sup>24</sup> Comeau RM, Fenster A, Peters TM: Intraoperative ultrasound in image-guided neurosurgery. *Radiographics* 18(4):1019-1027, 1998.
- <sup>25</sup> Vitek JL, Bakay RA, Hashimoto T, Kaneoke Y, Mewes K, Zhang JY, Rye D, Starr P, Baron M, Turner R, DeLong MR: Microelectrode-guided pallidotomy: technical approach and its application in medically intractable Parkinson's disease. *J Neurosurg* 88(6):1027-43, 1998.
- <sup>26</sup> Finnis KW, Starreveld YP, Parrent AG, Sadikot AF, Peters TM: 3D Functional database of subcortical structures for surgical guidance in image guided stereotactic neurosurgery. In *Proc. Of MICCAI '99*, volume LNCS 1679 pp758-767, 1999.
- <sup>27</sup> Finnis KW, Starreveld YP, Parrent AG, Peters TM: A 3-Dimensional database of functional anatomy, and its application to image-guided neurosurgery. In *Proc. Of MICCAI '00*, in press, 2000.
- <sup>28</sup> Horsley V, Clarke RH: The structure and functions of the cerebellum examined by a new method. *Brain* 31:45-124, 1908.
- <sup>29</sup> Roessler K, Ungersboeck K, Aichholzer M, Dietrich W, Goerzer H, Matula C, Czech T, Koos WT: Frameless stereotactic lesion contour-guided surgery using a computer-navigated microscope. *Surg Neurol* 49(3):282-288, 1998.
- <sup>30</sup> Alp MS, Dujovny M, Misra M, Charbel FT, Ausman JI: Head registration techniques for image-guided surgery. *Neurol Res* 20(1):31-37, 1998.
- <sup>31</sup> Laitinen LV, Liliequist B, Fagerlund M, Eriksson AT: An adapter for computer tomography-guided stereotaxis. *Surg Neurol* 23(6):559-566, 1985.
- <sup>32</sup> Taren J, Ross D, Lu Y, Harmon L: 3D laser scanning for image guided stereotactic neurosurgery. *Acta Neurochir Suppl (Wien)* 64:45-48, 1995.
- <sup>33</sup> Vinas FC, Zamorano L, Buciuic R, Li QH, Shamsa F, Jiang Z, Diaz FG: Application accuracy study of a semipermanent fiducial system for frameless stereotaxis. *Comput Aided Surg* 2(5):257-263, 1997
- <sup>34</sup> Hirschberg H, Kirkeby OJ: Interactive image directed neurosurgery: patient registration employing the Laitinen stereo-adapter. *Minim Invasive Neurosurg* 39(4):105-107, 1996.
- <sup>35</sup> Wada J, Rasmussen T: Intracarotid injection of sodium amytal for the lateralization of cerebral speech dominance: Experimental and clinical observations. *J. Neurosurg* 17:266-282, 1960
- <sup>36</sup> Penfield W, Jasper HH: *Epilepsy and the Functional Anatomy of the Human Brain*. Boston: Little, Brown, 1954.
- <sup>37</sup> Muzik O, Chugani DC, Shen C, da Silva EA, Shah J, Shah A, Canady A, Watson C, Chugani HT: Objective method for localization of cortical asymmetries using positron emission tomography to aid surgical resection of epileptic foci. *Comput Aided Surg* 3(2):74-82, 1998.

- 
- 38 Cosgrove GR, Buchbinder BR, Jiang H: Functional magnetic resonance imaging for intracranial navigation. *Neurosurg Clin Na Am* 7(2):313-322, 1996.
- 39 Fried I, Nenov VI, Ojemann SG, Woods RP: Functional MR and PET imaging of rolandic and visual cortices for neurosurgical planning. *J Neurosurg* 83(5):854-861, 1995.
- 40 Vitek JL, Bakay RA, Hashimoto T, Kaneoke Y, Mewes K, Zhang JY, Rye D, Starr P, Baron M, Turner R, DeLong MR: Microelectrode-guided pallidotomy: technical approach and its application in medically intractable Parkinson's disease. *J Neurosurg* 88(6):1027-43, 1998.
- 41 Vitek JL, Bakay RA, Hashimoto T, Kaneoke Y, Mewes K, Zhang JY, Rye D, Starr P, Baron M, Turner R, DeLong MR: Microelectrode-guided pallidotomy: technical approach and its application in medically intractable Parkinson's disease. *J Neurosurg* 88(6):1027-43, 1998.
- 42 Brown NJ, Britton KE, Plummer DL *Int J Med Inf* 1998 Feb;48(1-3):227-38, Standardisation in medical image management.
- 43 The 1998 DICOM draft standard, located online at <ftp://ftp.nema.org/> medical/dicom/final/
- 44 The National Electrical Manufacturers Association home page: <http://www.nema.org/nema/> medical/dicom/
- 45 Neelin, P: *The MINC Programmer's Guide*. Online at <http://www.bic.mni.mcgill.ca/> software/minc/
- 46 University Corporation for Atmospheric Research Unidata Program Center, Boulder, Colorado, USA
- 47 Schroeder W, Martin K, Lorensen B: *The Visualization Toolkit, 2<sup>nd</sup> Edition*. Upper Saddle River, NJ: Prentice Hall, 1997
- 48 The OpenGL website: <http://www.opengl.org>
- 49 The Python website: <http://www.python.org>
- 50 Salus PH: *Handbook of Programming Languages Volume 3: Little Languages and Tools*. Macmillan Publishing, 1998.
- 51 Northern Digital, Waterloo, Ontario, Canada
- 52 Horsley V, Clarke RH: The structure and functions of the cerebellum examined by a new method. *Brain* 31:45-124, 1908.
- 53 Roessler K, Ungersboeck K, Aichholzer M, Dietrich W, Goerzer H, Matula C, Czech T, Koos WT: Frameless stereotactic lesion contour-guided surgery using a computer-navigated microscope. *Surg Neurol* 49(3):282-288, 1998.
- 54 Alp MS, Dujovny M, Misra M, Charbel FT, Ausman JI: Head registration techniques for image-guided surgery. *Neurol Res* 20(1):31-37, 1998.
- 55 Laitinen LV, Liliequist B, Fagerlund M, Eriksson AT: An adapter for computer tomography-guided stereotaxis. *Surg Neurol* 23(6):559-566, 1985.
- 56 Taren J, Ross D, Lu Y, Harmon L: 3D laser scanning for image guided stereotactic neurosurgery. *Acta Neurochir Suppl (Wien)* 64:45-48, 1995.
- 57 Vinas FC, Zamorano L, Buciu R, Li QH, Shamsa F, Jiang Z, Diaz FG: Application accuracy study of a semipermanent fiducial system for frameless stereotaxis. *Comput Aided Surg* 2(5):257-263, 1997
- 58 Hirschberg H, Kirkeby OJ: Interactive image directed neurosurgery: patient registration employing the Laitinen stereo-adapter. *Minim Invasive Neurosurg* 39(4):105-107, 1996.
- 59 Blume WT, Parrent AG, Kaibara M: Stereotactic amygdalohippocampotomy and mesial temporal spikes. *Epilepsia* 38(8):930-936, 1997.
- 60 Hariz MI, Hirabayashi H: Is there a relationship between size and site of the stereotactic lesion and symptomatic results of pallidotomy and thalamotomy? *Stereotact Funct Neurosurg* 69(1-4 Pt 2):28-45, 1997.
- 61 Giller CA, Dewey RB, Ginsburg MI, Mendelsohn DB, Berk AM: Stereotactic pallidotomy and thalamotomy using individual variations of anatomic landmarks for localization. *Neurosurgery* 42(1):56-62, 1998.
- 62 Comeau RM, Sadikot AF, Fenster A, Peters TM: Intraoperative ultrasound for guidance and tissue-shift correction in image-guided neurosurgery. *Medical Physics* vol. 27, 787-800, 2000.