3D Slicer: A Platform for Subject-Specific Image Analysis, Visualization, and Clinical Support

19

Ron Kikinis, Steve D. Pieper, and Kirby G. Vosburgh

Introduction

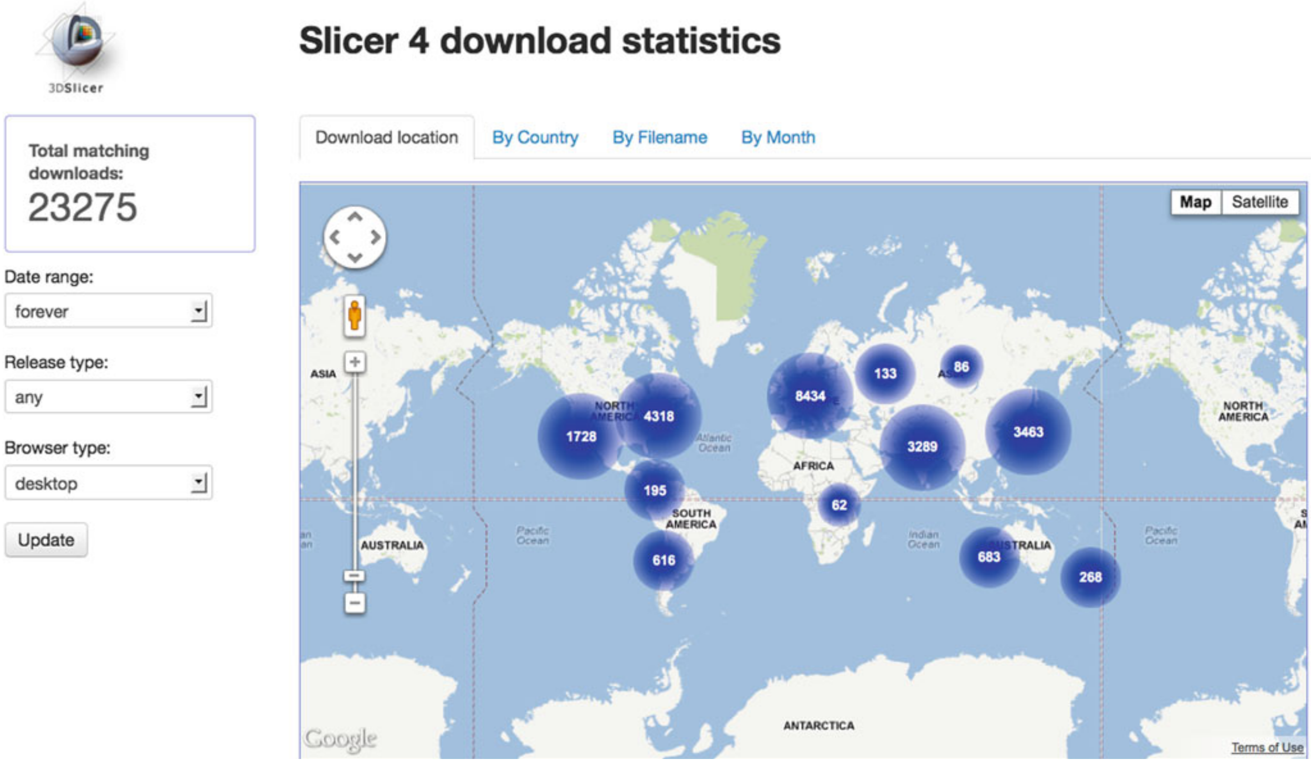
The continued evolution of the complexity and information richness of multidimensional data sets in medical imaging has spurred a parallel development of increasingly sophisticated and capable post-processing platforms for medical research as well as clinical care. Commercial imaging machines produce very high-quality 3D anatomic images and first-level displays of anatomic structures and some functional information, such as blood flow. Post-processing is required, however, for more advanced displays such as dynamic contrast imaging and diffusion tensor MRI. The impact of advanced medical imaging today is the result of three concurrent, interlinked revolutionary technological developments: (1) anatomically correlated digital data acquisition from the body, (2) advanced multistage processing platforms to extract and enhance information of relevance for research and clinical use, and (3) the continuing evolution of advanced computer science techniques to organize and sustain very high-quality platforms for data processing:

1. *Data Acquisition*: In 1970, computed tomography (CT) demonstrated that digital radiologic acquisition, image processing, and display could provide new information to guide care and enable research. CT and then magnetic resonance imaging (MRI) systems showed obvious clinical benefit by providing higher contrast tissue differentiation (at the cost of spatial resolution) and nonphysical displays (initially, axial slices of the head). The implementation of picture archiving and communications systems (PACS) supported and drove the conversion of medical imaging from a film-based reading and archival data storage system to a digital system. This was solidified in the 1990s, when it was established that electronic displays could be used without the loss of diagnostic quality and essentially completed by the widespread adoption of digital mammography.
2. *Post-processing*: In parallel with this evolution in digital imaging and infrastructure, basic, applied, and clinical researchers sought to improve the utility of these digital image data sets through image processing techniques to adjust contrast, reduce noise, and align related images for comparison and similar activities. In part, this effort was "pulled" by potential clinical benefits; in part it was "pushed" by the radiologists' need to have the complexity of these increasingly large data sets reduced so they could be processed quickly. The ability to make 3-dimensional visualizations of these inherently volumetric image data sets also developed in the mid-1970s but was hampered by limits on computing power and also the lack of sufficiently powerful algorithmic approaches and software capabilities. Radiology research communities began, in this period, to build vendor and self-organized working communities to share technology, define scientific approaches, and develop standards. Frequently these efforts were instantiated in commercial or academic site-supported software/hardware packages, which might be proprietary or open-source (Unix platform) based.
3. *Platform Development*: To fit into research and clinical timelines, straightforward post-processing is often integrated into the acquisition protocol with minimal user involvement. For clinical applications, this post-processing function becomes part of the system operation and must then meet the same standards of reliability, robustness, and simplicity of operation as the core imaging equipment. In many cases, this is accomplished using software packages added onto imaging system by the major vendors of medical

R. Kikinis, MD (✉) • K.G. Vosburgh, PhD
National Center for Image Guided Therapy,
Department of Radiology, Brigham and Women's Hospital,
Harvard Medical School, Boston, MA, USA
e-mail: kikinis@bwh.harvard.edu

S.D. Pieper, PhD
Surgical Planning Laboratory, Isomics Incorporated,
Cambridge, MA, USA

F.A. Jolesz (ed.), *Intraoperative Imaging and Image-Guided Therapy*,
DOI 10.1007/978-1-4614-7657-3_19, © Springer Science+Business Media New York 2014


Slicer 4 download statistics

Total matching downloads:
23275

Date range: forever

Release type: any

Browser type: desktop

Update

Fig. 19.1 This figure slows the geographic distribution and cumulative number of partial and full downloads of Slicer 4 in early spring 2012. 3D Slicer has a large number of active, committed users around the world

imaging equipment and many specialized companies. Examples include the General Electric Advantage Windows [1], Siemens Syngo Via [2], Vital Images (Toshiba) Vitrea [3], and the Visage Imaging Amira [4] packages.

As well, Analyze [5], developed by Richard Robb and his team at the Mayo Clinic, is an excellent general-purpose platform which features complex visualization and analysis capabilities using parallel processing of multiple image data streams. A more recent example is OsiriX [6], developed by Osman Ratib and his team, which has a hybrid strategy employing some closed and some open software elements. However, none of these existing packages are fully free and open source, meaning they can be used without restrictions and they are generally not available on multiple platforms. Such closed or semi-proprietary systems offer advantages for the capable and dedicated research user, but they may be less accommodating than open-source approaches to advancements in the base technology or adapted to the developmental changes required for broad clinical impact. Also, they lack the seminal advantage of an open-source approach, which is participation in a large community with varied interests, willingness to share strategies and techniques, and motivations to develop and use new capabilities.

A key step in the evolution of today's flexible and sophisticated capabilities for image-data-based research medicine was the creation of the 3D Slicer [7], also known as "Slicer" platform, which is based on a modular architecture [8]. 3D Slicer has become a successful and long-lived platform for the effective use of volumetric images in clinical research and procedure development. At first, 3D Slicer primarily supported the radiology research community. That effort was expanded in the past decade, particularly with support from the National Institutes of Health (NIH) through the Visible Human Program, the National Centers for Biomedical Computing (which supports the National Alliance for Medical Image Computing, NA-MIC), and the National Center for Image-Guided Therapy (NCIGT) to serve as resources for research efforts using radiologic data sets and related digitized volumetric data.

Slicer's success, as measured by software downloads by users around the world (Fig. 19.1), has been created by an open international developer community with processes in place to enable engagement of participants, a software engineering methodology to support multisite development and maintain stability, an active, open, international user community, and the effective formalism of an open-source license model. From the perspective of a user, Slicer then has these principal attributes:

- The use of an open-source strategy based on a curated and continually tested software object library. This means not just that Slicer-based software is fully transparent to the end user but also that its robustness and reliability are freely inspectable and testable.

- A vibrant and growing user community, which supports Slicer's core principles and works to extend Slicer capability to serve the evolving needs of the research and clinical users.
- A core belief that an untrained user who is not a computer scientist should be able to use the system successfully with appropriate training. This differentiates Slicer from many very effective and useful platforms for advanced image data processing, which must be operated by experts, generally for research purposes which may use specialized computing architectures or high-bandwidth communications.
- Slicer is supported on Windows, Macintosh, and Linux. It can be run effectively on commodity personal computers and is easily embedded in larger systems for, for example, real-time data acquisition and processing.

3D Slicer Technology

3D Slicer is a free open-source software platform distributed under a BSD-style license for the analysis and visualization of biomedical images. It is modular and extensible and used for research and application development in medical diagnosis, staging, and therapy guidance.

3D Slicer supports several different plug-in mechanisms, with a variety of complexities and capabilities. Correspondingly, platform developers may work at any level of the system to implement changes. Slicer is available for Windows, MAC OSX, and Linux operating systems and built on VTK [9], ITK [10], and other packages comprising the NA-MIC Kit [11]. All Slicer versions from 4.0 on provide native 64-bit capability. 3D Slicer then consists of an assembly of compatible, modular toolkits, tools, and methodologies. Documentation and tutorials may be downloaded from the Slicer website [7], which also provides information about a variety of uses.

The users of Slicer span a range of interests and capabilities. At one extreme are users who desire a stable and robust platform for basic image analysis and display functions but wish to use certain features of Slicer, such as the ability to know exactly how the data are processed, the ability to add simple upgrades and improved functions as they may be useful, and the participation in the Slicer community. Users with more complex needs may wish to modify program operations, add specialized processing or interfaces, or even re-architect the overall system. Slicer has evolved to accommodate this range of users by implementing an increasingly sophisticated support infrastructure while adhering to its core principles (Fig. 19.2).

The components of a Slicer session are organized into the "MRML tree" [12]. The MRML tree is organized hierarchically with nodes, which contain such content as volumes, 3D models, and transforms. Nodes also store window level, layout parameters, and related factors. "Sceneviews" capture the complete MRML tree, encompassing the build of Slicer code, including layouts, volumes in the slice viewers and their visibilities, and the 3D viewers. For example, distinctions are made between objects which are made visible (labeled with a specific color) or which is made transparent. Sceneviews offer a convenient process to capture complex configurations for reviews, meetings, presentations, or review of a clinical therapy plan. While Slicer itself is written as an interactive program for subject-specific analysis, many of the Slicer modules can be run from the command line so they may be used in processing pipelines for offline automated analysis of larger studies. Slicer can also be applied to the prototyping of workflows. A first generation of workflow engines is being developed to support this use case [13].

The successful operation of Slicer depends on a process infrastructure to ensure robust and repeatable computational performance. A number of specialized software tools were developed and are employed for this purpose: git, a web-based source code repository [14], manages contributed source code; cmake [15] manages compilation of the source code on a variety of operating system platforms. Automated testing is performed on each platform every day after the source code has been compiled. This is accomplished by a software package called ctest [15]. Cdash [16] is a web-based interface that presents the results of the nightly compile/test cycle on all the platforms in which Slicer has been compiled. With several platforms (Windows, Linux, Mac) and several hundred tests per compilations, cdash provides an interactive interface for managing the complexity underlying this approach. This system enforces the key Slicer requirement that every element of code in the Slicer library is fully functional, compatible with the other code elements, and accurate.

New stable releases of Slicer occur biannually. As of early 2012, the stable release is 4.1. Compared to its predecessor, key functional features of this release include improved data processing and rendering speeds and a more intuitive and powerful user interface toolkit. Slicer 4.1 has a simpler user interface, more flexibility in configuring the layout with multiple viewers, a more robust volume-rendering module, integral charting abilities, and greatly improved support for DICOM. Under the hood, the GUI was migrated to Qt, and the source code tree was reorganized and pruned. Support for Python, a powerful and popular scripting language, was strengthened. Significantly improved support for web technologies is possible through WebKit, a web browser engine which is available as part of Qt. Slicer 4 also relies on CTK, a new toolkit for UI components. Particular new capabilities relevant to image-guided therapy are the ability to specify Slicer operating modes in terms of a procedure workflow and improved interfaces with external devices. Finally, the under-

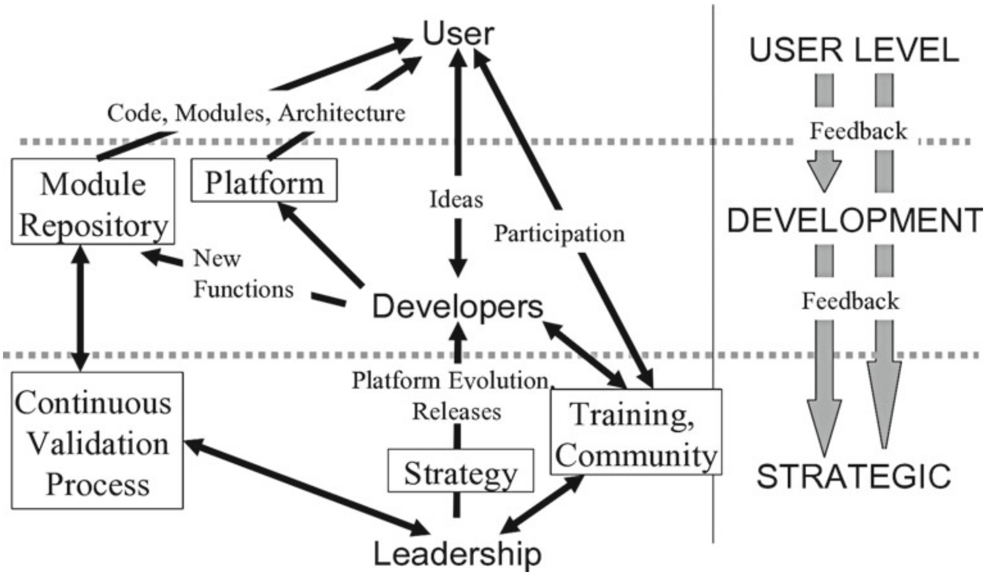


Fig. 19.2 The main elements and working relationships of the 3D Slicer community. The user accesses training and support from the Slicer community and downloads the basic software package and extension modules appropriate to his or her task. Developers create new functions which may be included in the module repository for any user download. The leadership sets strategy (architecture, new release planning, etc.), maintains the continuous validation process which validates elements of the code repository, and organizes training and community events such as "Project Weeks." Openness and feedback are primary characteristics of the Slicer community

lying code has been reorganized and streamlined for greater performance.

Using Slicer

3D Slicer [7], is a software platform which continually evolves to serve the requirements of end users and computer scientists. While Slicer began several years ago as a master's thesis project at MIT [17], today it is a widely used platform for scientific and clinical medical image processing applications.

We distinguish two communities supported by the Slicer platform:

- *General Users*. Use Slicer as a tool to conduct basic, clinical, and translational research and clinical applications. A typical task might consist of loading a data set and processing it using multiple segmentation algorithms, then comparing and displaying the results. Key attributes for this medical-user community are ease of use, adaptability, support for translation to clinical demonstrations and technology transfer, and robust algorithms that accommodate anatomic and pathologic variations. Many of the supporting core technical capabilities are also effective in nonmedical applications; users work on the Slicer platform in fields ranging from 3D design to complex geology data reduction, often adding extensions to fit their purposes. Slicer is often used at the investigational stage of clinical validation of a new research, diagnostic, or therapeutic approach. Because 3D Slicer-based systems are both robust and flexible, they enable investigators to rapidly implement and prove out new concepts, explore options, and gather data for subsequent analysis.
- *Computer Science Users*. A worldwide open community of software developers and scientists is constantly extending and improving 3D Slicer. A web-based software engineering methodology has been used and evolved to ensure that each individual's contribution is respected and also that the integrated total capability maintains functionality and transparency.

Running Slicer as a Workstation

3D Slicer was initially intended (and is widely used today) as a workstation with static data sources. Slicer can run directly on a conventional desktop or laptop. Data is loaded, analyzed, and displayed, and the task (such as a diagnostic analysis, image biomarker quantification, or the planning of a surgical procedure) is completed.

When beginning a new project, the user configures and runs an initial build of standard Slicer modules from the curated Slicer library [8]. Nonstandard modules ("extensions") may also be loaded. The resultant "work platform" will include an overall workflow and modular elements for I/O, data preprocessing, data processing, subsequent analysis and modification (such as 3D model construction), and, at several of these stages, display. The operator controls the progress through the workflow, selects options, and inputs parameters with the GUI and keyboard. All processing is local to the end user. Trust issues associated with transmission of confidential patient information on a nonlocal infrastructure such as the Internet or “cloud” server are thus avoided.

Typically Slicer is used as a package, but it may be adapted to the user’s purposes using an interactive editing capability. The user need load only the modules relevant to his or her task, so the hardware is used efficiently. To work with modern diagnostic images, computers must have significant amounts of memory and a powerful graphics accelerator board. Here we describe some of the primary elements of a 3D Slicer build.

Graphical User Interface (GUI)

Slicer has a flexible layout with a panel for shortcuts at the top and a wide panel on the side that is used for displaying module-specific information. These modules provide access to the majority of the specific functions that control most of the processing (see Fig. 19.3).

Data Input

Slicer can load data in a variety of image formats, including DICOM [18], with the number of volumes loaded limited only by the memory available to the program. Slicer provides a GUI to import DICOM data from disk and query/retrieve data from a PACS system and, in addition, can add a variety of data formats popular in research, including NIFTI, NRRD, TIFF, and raw raster data. Slicer has integral support for a number of more specialized formats including label maps, tensor data, triangulated surface models, and diffusion-weighted images. Volumes can be associated with “look-up tables” (LUT) that can be used to assign colors and names to particular values in the volume. Slicer comes with a number of built-in LUTs and can also load customized LUTs. See also the discussion below regarding use of Slicer with real-time data.

Display

Images are typically displayed in 2D viewers which display three volumes: foreground, background and a label map, and a variety of combinations of blending and overlays of the three selected volumes. The images can be zoomed and panned. The contents of each of the 2D viewers can also be displayed in 3D viewers that can display triangulated surface models, cross-sectional slices, volume renderings, and a number of other specialized graphics such as glyphs for

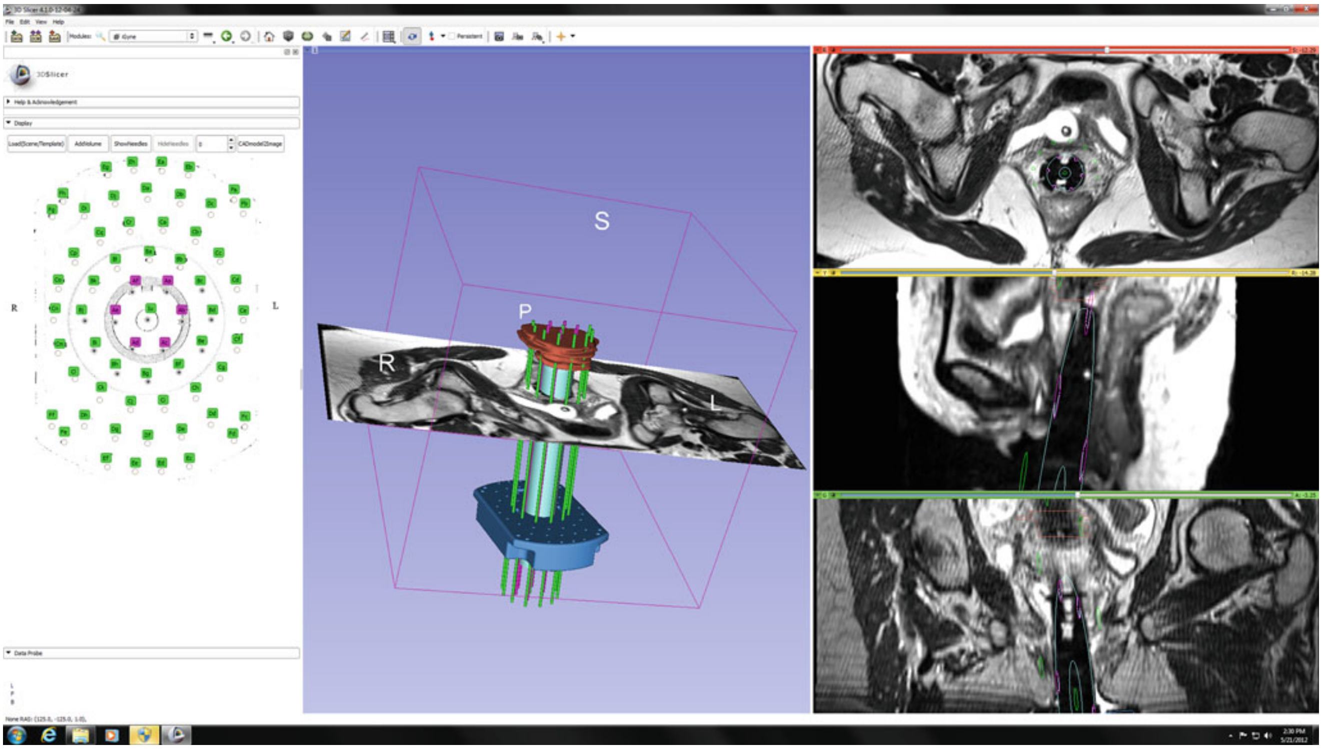


Fig. 19.3 Typical Slicer 4 implementation, which shows an intraprocedural planning display, made during MR-guided cervical cancer brachytherapy. On the *left* is the template to guide the brachytherapy insertion needles, showing the channels in use. The *center* shows a 3D rendering of the needle guide system around the central obturator, with a registered axial MR image which may be positioned as desired by the physician. The axial, sagittal, and coronal images are shown on the *right*; these may be located independently to optimize visualization of the tumor and its relationship to the therapeutic needle system (Credit, X. Chen, et al., see Chap. 34)

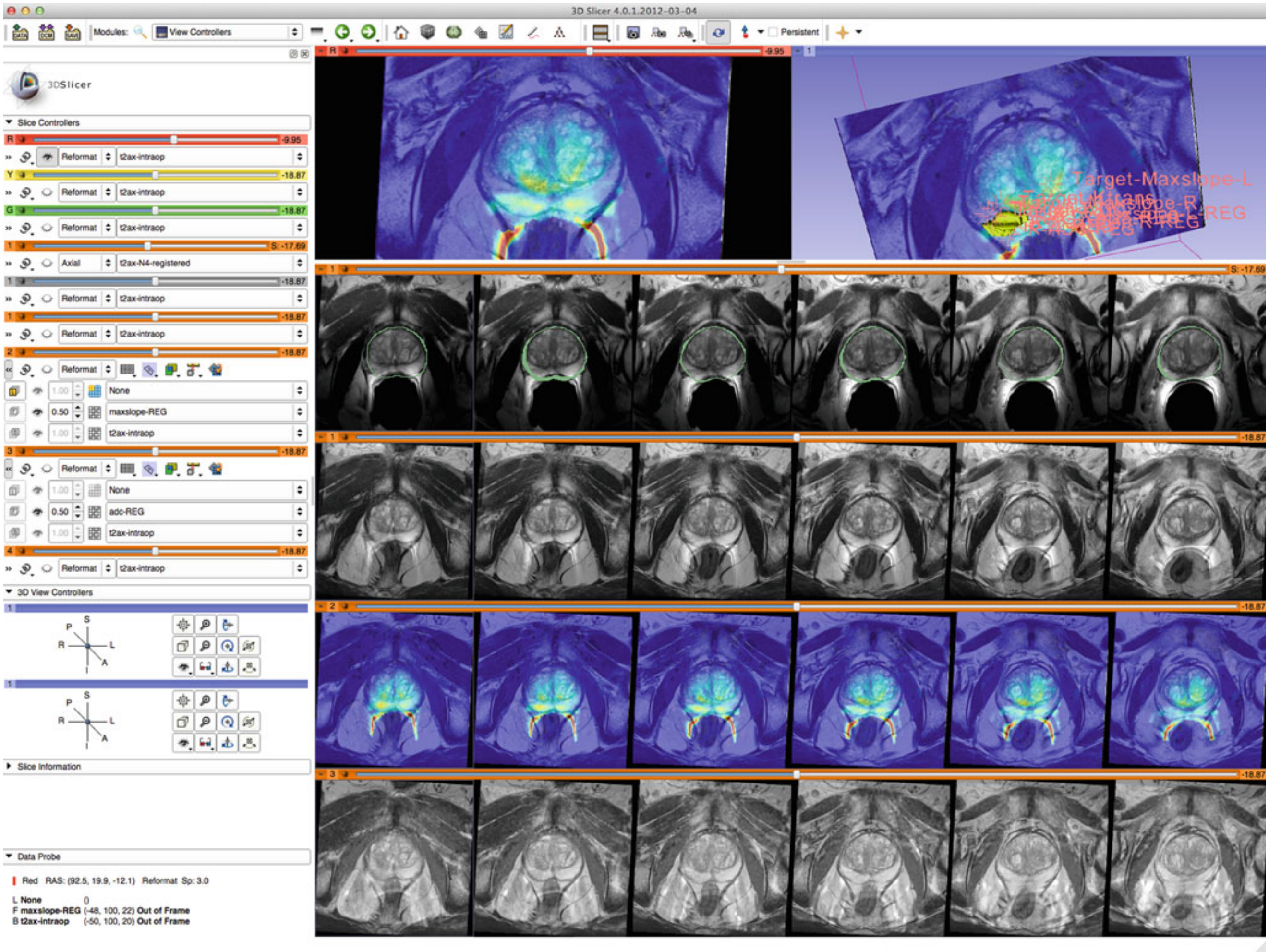


Fig. 19.4 These images from prostate therapy planning show the power of 3D Slicer in displaying multiple images using the “compareview” layout

representing tensors and scalar fields associated with surface models (e.g., results of processing of brain images with www.freesurfer.net). In the 3D viewers, users can combine generated 2D and 3D displays. Also available are “compare” viewers (Fig. 19.4) that are configure to compare different images whereby a number of slices are arranged in a horizontal or vertical array. The layout manager provides a number of predefined configurations with combinations of 2D and 3D viewers.

Core Functions: Segmentation, Registration, and Volume Rendering

Segmentation

Segmentation is the process of labeling each image element in a data set to permit the grouping of like elements into regions or volumes. Thus, if a tumor appears dark in a data voxel, all dark elements might be labeled “tumor” and made, for example, a different color when the image is displayed. Slicer’s interactive framework includes the Editor described below and a number of semiautomated algorithms such as region growing [19] and fast marching [20]. The automated algorithms include a comprehensive EM segmentation framework and some level-set capabilities focused on segmentation of tubular structures. The EM segmenter can be used for segmentation of a number of objectives. A library of sample solutions provides ready-to-use solutions and starting points for modifications (see [21] for more information). VMTK, a processing package focused on segmentation and analysis of vessels, is available as an extension to Slicer. The modules contain tube-sensitive filters (for preprocessing) and level sets for the actual segmentation [22]. Atlas data libraries may also be loaded (Fig. 19.5).

The Interactive Editor [23] is a framework for interactive segmentation of volumetric scalar data that contains a variety of tools called “effects.” Effects include a variety of capabilities from painting, polygonal outlining, and thresholding through many connectivity and morphology tools to a novel competitive region-growing algorithm called GrowCut [24] (see Fig. 19.6). Typically, a grayscale volume such as a structural MRI is used for the segmentation, and the results of the process are deposited in a "label map." An informatics capability has been integrated into the Editor to allow it to split, edit, and merge multiple labels. A specialized module, ModelMaker, establishes a pipeline for generating multiple surface models from a label map. Names and colors are inherited from the label map LUT, and parameters allow for tuning the extent of smoothing and triangle reduction performed on the models [25]. The 3D extent of segmented structures can be used as a starting point for further analysis. For example, the label statistics module can be used to calculate the volumes of organs or tumors or to assess the distribution of signal intensities of a grayscale image with a user-defined region of interest.

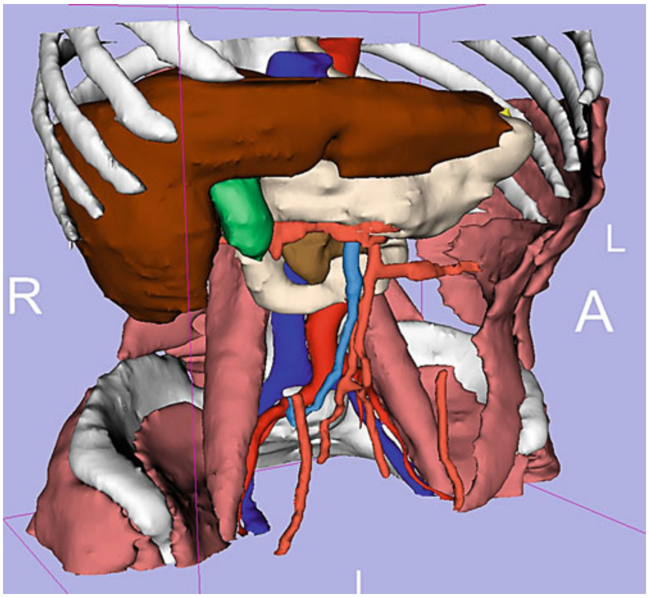


Fig. 19.5 Anatomic atlases play an important role as prior data to improve the speed of advanced segmentation. This abdominal atlas was created by the Slicer community and is available for free download

Annotation may be applied to images or to volumetric models. Text annotations, rulers, and a number of other elements can be used to annotate structures of interest, such as brain structure measured by diffusion MRI (Fig. 19.7). In combination with the Sceneviews, this turns Slicer into a very powerful tool for reporting findings on complex data sets. Further examples are the used of ellipsoids for diffusion tensor imaging display (Fig. 19.8) and glyphs to represent instruments (Fig. 19.9).

dMRI Processing

Diffusion MRI (dMRI) processing is a mature capability of Slicer [26]. In contrast to many other dMRI packages, which are designed for group comparisons in the context of basic neuroscience research, Slicer is focused on subject-specific analysis. Since data representing diffusion-weighted images are not yet provided in a stable, DICOM-compliant format, Slicer tracks and adapts to minimize this issue: the DICOM to NRRD converter [27] is constantly updated and supports a large number of formats. Noise reduction filtering [28] and tensor estimation modules [29] are available and execute within a few minutes on regular-sized data sets. Sophisticated single-tensor streamline extraction is possible using label maps, surface models, and fiducials as seeds (see [26] for more information). Interactive visualization of streamlines provides a number of different visualization capabilities. Tractography input points can also be driven by the locations of tracked surgical instruments using the OpenIGTLink interface described below.

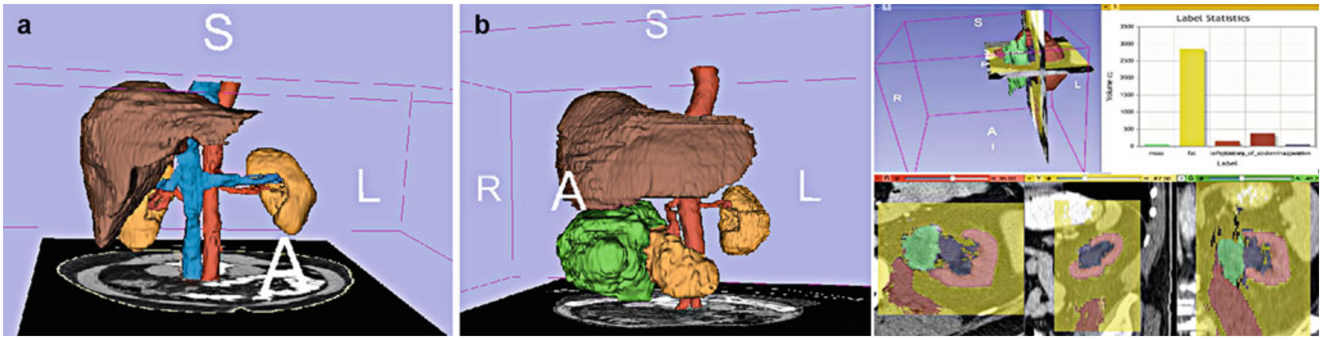
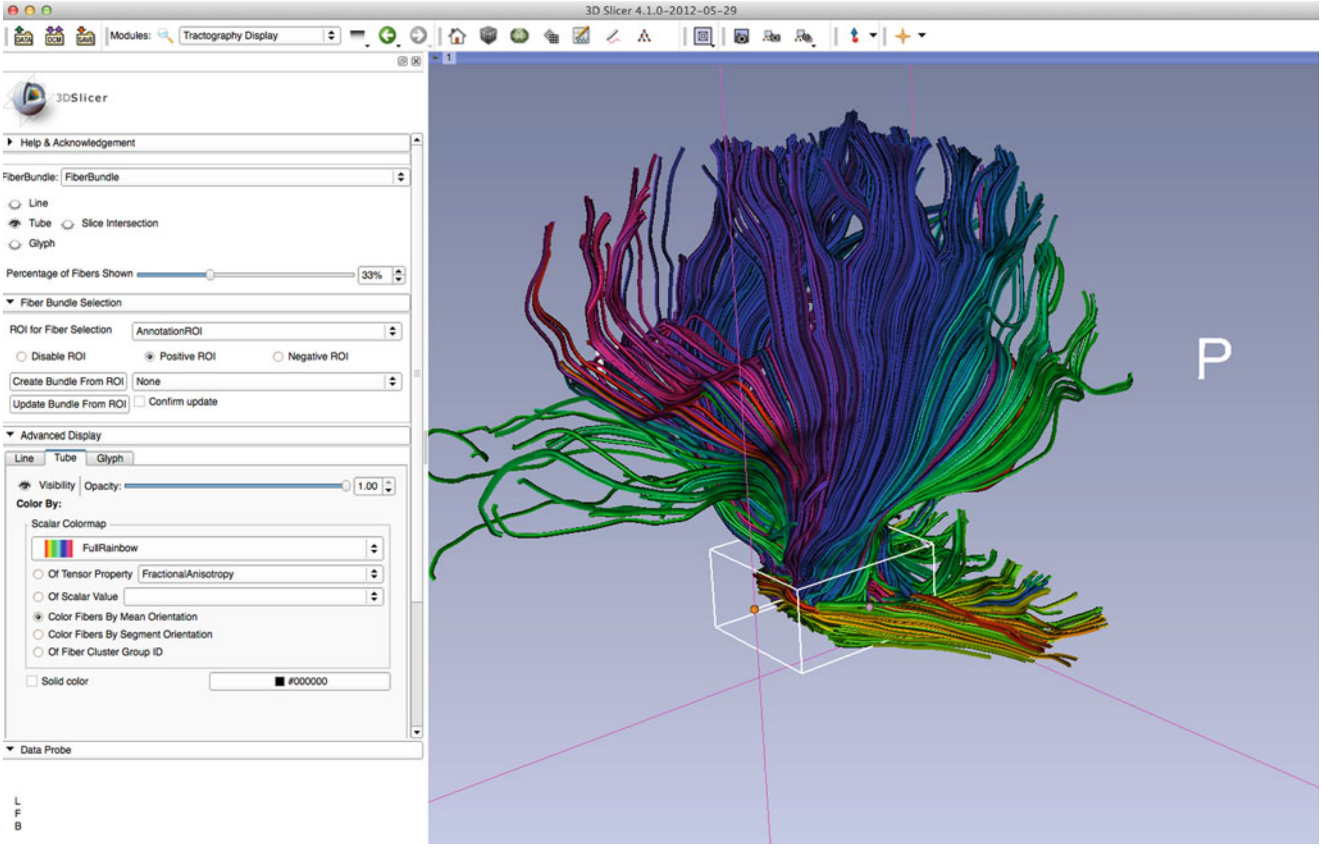


Fig. 19.6 Surgical planning for surgery of an abdominal sarcoma. Figure (**a**) shows normal anatomy of the liver, kidneys, and major vessels. In (**b**), the anatomy of a patient with a malignant mass (*green*), which is displacing the right kidney. The *right panel* shows a near real-time segmentation of the tumor and kidney which could be performed during the surgical procedure. The goal is to display the extent of invasion of the tumor into the kidney parenchyma and vasculature



Modules: Tractography Display

FiberBundle: FiberBundle

○ Line
● Tube ○ Slice Intersection
○ Glyph

Percentage of Fibers Shown 33%

ROI for Fiber Selection AnnotationROI

○ Disable ROI ● Positive ROI ○ Negative ROI

Create Bundle From ROI None

□ Confirm update

Visibility Opacity: 1.00

FullRainbow

○ Of Tensor Property FractionalAnisotropy

○ Of Scalar Value

● Color Fibers By Mean Orientation

○ Color Fibers By Segment Orientation

○ Of Fiber Cluster Group ID

□ Solid color ■ #000000

Fig. 19.7 Color-coded tracts segmented from diffusion tensor MRI to support surgical planning and intervention

Registration

Registration is the process of fitting data acquired from an object to another data set which is gathered from the same object under different conditions. The coordinate system infrastructure provides the foundation upon which registration is accomplished. While the data are kept in their original voxel space, all objects in Slicer share the same Cartesian anatomical coordinate system called RAS (right anterior superior). Data in voxel space is transformed into RAS space before being displayed (see [30] for more information). Additional transforms can be introduced interactively or programmatically. The Transforms module allows interacting with a transform in several ways. Elements of the MRML tree can be pulled under a transform and transforms can be combined. Interactive rigid registration is often useful as an initialization step for more sophisticated (and less robust) higher-order algorithms. In addition to the interactive registration, Slicer has a comprehensive framework of data-driven automated pipelines for rigid and higher-order registration, including B-spline and demons-based nonrigid algorithms. The iterative closest point (ICP) algorithm is available as a way to register models and point clouds. At a higher processing level, Brainsfit [31], a pipeline based on ITK components, uses mutual information to rapidly and robustly register multiple data volumes. Originally developed, as its name implies, for brain image analysis, it is applicable to many other volume data acquisition series. Resampling tools allow the application of a transform to other data sets, to guide the user in this complex process. Slicer provides a library of examples consisting of a sample data repository and matched solution workflows (see [32] for more information).

Volume Rendering

This core module supports ray-cast volume rendering of a volume [33] by tracing virtual rays through the volume along lines defined by the eyepoint and the pixels on the display screen. Volume rendering is helpful for initial data exploration and as a component of complex visualization. Since it is computationally demanding, volume rendering often requires hardware acceleration. It is difficult to design a generic volume-rendering engine that meets all needs; thus today Slicer provides four volume-rendering engines and supports three different GPU accelerator cards, with the configuration being determined by the hardware availability and

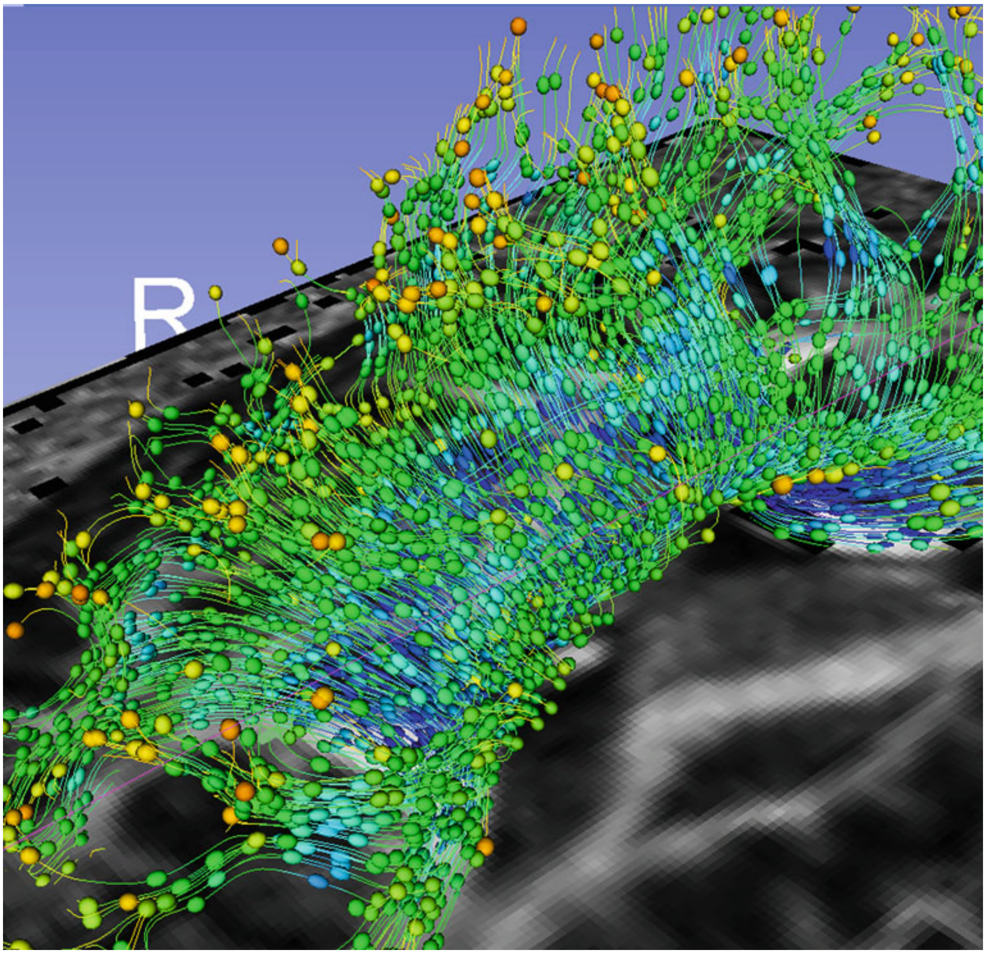


Fig. 19.8 Diffusion tensor streamline tracts augmented with ellipsoid glyphs to show the degree and extent of anisotropy in the brain tissue

the operating system. The appearance of data during volume rendering is determined by "transfer functions." Interfaces allow interactive modification of transfer functions to adjust them to individual data sets. A number of predefined transfer functions are available in Slicer; also, users can create, store, and restore their own transfer functions. Volume rendering is particularly popular for CT data, where predefined transfer functions work well, since CT signal intensities are less variable than for other modalities. Slicer volume rendering allows for regions of interest to crop the volume used for visualization. Some of the render engines support compositing the volume rendering with other visualization elements such as cross sections and triangulated surface models.

Image-Guided Therapy (IGT) Capabilities

3D Slicer and other image post-processing systems were first used offline, since the computational speed and data management capabilities of the original systems were too limited to fit into the clinical workflow. However, the benefits of "closing the loop" to guide and measure the effects of interventional therapies were evident from the beginning. In addition to higher speed and data capacity (which were supplied quickly by commercial computer hardware development), bidirectional data transfer techniques were developed to refresh and augment imaging data, new approaches were developed for spatial calibration and registration (to link the real and virtual model spaces), and tracking systems were created to measure positions of anatomic features, fiducial markers, and instruments. Over time, these capabilities were stabilized in standards such as OpenIGTLink [34] is a bidirectional communication protocol which allows the system developed to interface imaging and other data in near real time. OpenIGTLink is supported by a number of vendors including NDI and Brainlab and has been demonstrated in use with robots and scanner control.

Examples of Workstation Use

From its inception, Slicer has been used to plan and characterize the effects of therapy. As well, it has evolved from a data processing platform into a real-time operating environment for image analysis and further into a programmable

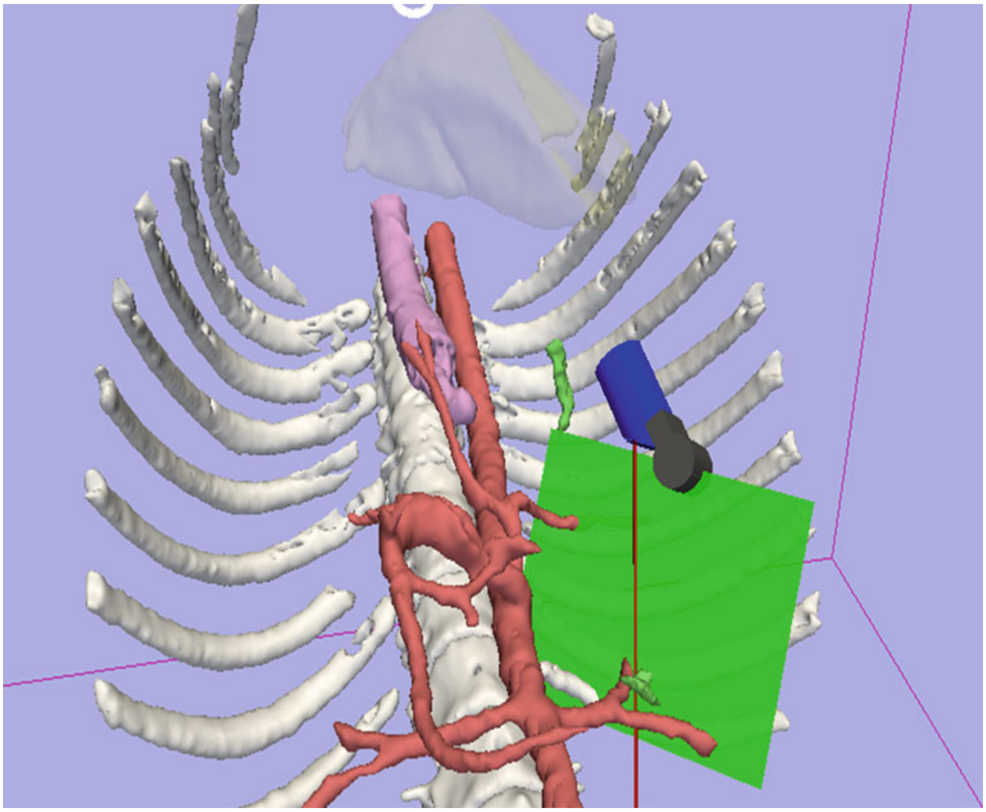


Fig. 19.9 Use of 3D Slicer as a real-time display system for procedure guidance (see Chap. 39). The *blue* and *grey* glyph shows the position and orientation of an endoscopic ultrasounds probe in the stomach of a porcine model. The *green rectangle* glyph displays the plan of the ultrasound transducer; the *red line* is the nominal track of the integrated biopsy needle

open-loop control environment for direct use in image-guided medical procedures and other real-time applications. As technology advancement has permitted closer integration of pre-procedure and intraprocedural images with complex 3D models and treatment plans, the Slicer platform and its capabilities have been enhanced to provide more timely and useful data to clinicians.

Many chapters in this book describe applications of Slicer used as a workstation in support of clinical care. While widely used in neurosurgery [35], applications encompass the entire body. Typical examples for the uses of Slicer in surgical planning include Fig. 19.6a, b and examples shown in Chaps. 7, 19, and 42. As well, Slicer is used for intraprocedural modeling of radiation therapy positioning and brachytherapy dosage; Chaps. 32 and 39 discuss the deformation of anatomy during the procedure (brain sag, shifts due to position changes). Also, Slicer is widely used for research applications, such as studies of brain injury [36, 37].

Real-Time Use of Slicer: Closing the Loop for Therapy Control

Slicer has several real-time capabilities to support image-guided therapy, including an interface for bidirectional communication with devices such as tracking systems, robots, and imaging devices: OpenIGTLink. For example, OpenIGTLink can drive a transformation matrix that can move objects such as 3D models of a probe. OpenIGTLink can also be used to prescribe a scan plane for acquisition on a scanner from within Slicer. The Slicer workflow module will track progress in a procedure in real time and implement appropriate analysis and displays to guide the clinician. For examples, see Chap. 42 for the use of Slicer in neurosurgery planning and Chap. 32 for image-guided endoscopy.

Working with Slicer as a Developer

A primary advantage of an open-source platform is that it allows users and developers who are not part of the developer team to modify the program to their liking and in this way expand the range of applications and introduce new capabilities. 3D Slicer is a complex platform dependent on a large number of toolkits and libraries and compiled on several host systems. The core development team is distributed as well. This situation has required the adoption of a software engineering methodology that includes automated testing, nightly compiles on all supported platforms, and a sophisticated source code repository [14, 38]. This infrastructure is supported by a substantial investment in software engineering, particularly to ensure platform scalability.

Scripts and documentation have been developed to lower the barrier of entry and understanding for new developers. In addition, a number of plug-in mechanisms have been developed to enable nonexperts to write simple extensions to Slicer's capabilities or to integrate their stand-alone programs into Slicer. Some of these are easy to learn and require only basic knowledge of programming, while others provide full control over the majority of Slicer's capabilities but require in-depth programming knowledge and skills. Potential developers can choose between those two approaches depending on their needs. The command line interface (CLI) is the simplest plug-in; it can be learned in a few hours. Tutorials for adding ITK and Python-based extensions are available. "Loadable plug-ins" provide more control over Slicer but require more knowledge about Slicer. To support and build the Slicer developer community, a working event called Project Week is held every 6 months and is open to everyone interested (http://www.na-mic.org/Wiki/index.php/Engineering:Programming_Events).

The version of Slicer distributed through the website www.slicer.com contains only open-source code without license restrictions, which also means there is no known inclusion of patented methods. Documentation and some automated tests are also supplied for all tools in the official distribution. To accommodate modules that are less mature, do not have tests, or are distributed under more restrictive licenses, an informatics infrastructure has been developed to support externally contributed plug-ins, i.e., extensions [39]. The main requirement for such extensions is a valid email address for contacting the author and an explicit declaration of the licensing terms for code distribution. In the future, new modules will primarily be added through this extension process.

3D Slicer as a Commercial Development Platform

Development using open-source software systems can be an attractive approach for commercial use when complex functions are needed in products with comparatively low costs and volumes. Proprietary software systems often have excellent performance, but they can be expensive to create and difficult to maintain. Thus they build significant costs into the products. Considering these aspects in more detail, using 3D Slicer as an example:

1. Building and maintaining a complex embedded software system is expensive and difficult to manage. In particular, it is hard to attract first-rate talent to commit to this type of internal development in an environment where they lack professional peers and outlets to share their skills or get credit.
2. Validation and training eat up resources, particularly as the number of platforms, use cases, and geographical sites expand.
3. As time passes, commitment to a proprietary approach becomes more difficult. The team and the business become wedded to hardware and software platforms, which are expensive to change, but whose performance deteriorates relative to the ongoing work in the field.
4. The work cannot reasonably be protected by patents, so it must be closely held as art, adding to the isolation of the software team.

In contrast, the use of open-source software, such as 3D Slicer, gives several advantages:

1. Economies of scale are inherent. A system like Slicer has thousands of infrastructure developers competing to keep everything up to date and performing correctly.
2. Slicer accommodates many of the issues associated with hardware and software platform evolution and changes.
3. The dedicated staff can be much smaller, because so much of the code is simply uploaded, and they spend much less time debugging because the Slicer code is heavily tested.
4. As long as the staff participates fully in the open-source community, they can latch onto the best and brightest ideas for the core of product functionality.
5. Also, they acquire a peer group automatically. Particularly for isolated software groups, this external community is a very helpful balance to the internal corporate (or laboratory) culture.
6. The key algorithms and approaches ("crown jewels") need not be shared. If, for example, 3D Slicer is being used, such proprietary functions can be built into the code as extensions.

Given the overall evolution of the medical imaging processing field, an open-source approach such as 3D Slicer may itself become obsolete. For example, commercially available systems may provide an adequate solution to the user's need and be simpler to operate. If such systems are FDA approved, that is a key factor in expanding the space of clinical use. Slicer-based systems may be FDA approved, but then they become essentially as "closed" as any other approved software system.

Training and Documentation

Training and documentation are integral to Slicer; they begin with tooltips that appear on rollover of most elements of the user interface. Each module has a help and acknowledgement tab with onboard help. The Slicer website contains more comprehensive documentation. To enable self-training, a rich portfolio of tutorials with matched data sets is available online [40]. To complement this web-based resource, hands-on training events ranging from half-day tutorials to multiday hands-on workshops are offered in a number of different venues, including conferences and at different universities. With support from the NIH and collaborating projects, nearly 1,000 researchers attended such workshops in 2010.

Summary

3D Slicer is a powerful free software platform that is easy to use, extensible, and available on all the major operating systems on a user's computer and behind a firewall. The combination of open source that allows inspection of the code and the local hosting of the software reduces concerns about confidentiality of patient identifiers in a clinical research setting.

Remember that Slicer is not FDA approved and does require the user to comply with any specific regulatory requirements and institutional roles regarding research and clinical use. As an end-to-end open solution for several image-guided therapy research and demonstration domains, Slicer is uniquely flexible and powerful. It provides a reference implementation and proof-of-concept for many new and early stage technologies. Developers of commercial clinical systems are encouraged to draw inspiration or even freely copy entire blocks of code into their systems for routine clinical deployment. The Slicer project has been pursued with the goal of accelerating and transitioning novel image analysis and image-guidance technologies into practical tools for improving human health.

Conclusion

A free, open-source, and easily extensible software package, 3D Slicer provides a powerful, easy to use, image informatics framework for visualization, segmentation, and registration for the research, development, and transitional phases of imaging-based research and image-guided therapy.

Acknowledgments The authors thank the 3D Slicer development community members for their efforts and recognize with gratitude the support of the NIH through NAMIC, NAC, NCIGT, and portions of many other projects.

The authors were supported by the Neuroimage Analysis Center (NAC), an NIBIB Resource Center, NIH NIBIB P41 grant EB015902, and the National Alliance for Medical Image Computing (NA-MIC), funded by the National Institutes of Health through the NIH Roadmap for Medical Research, Grant U54 EB005419. Information on the National Centers for Biomedical Computing can be obtained from http://nihroadmap.nih.gov/bioinformatics. Drs. Kikinis and Vosburgh were supported in part by the National Center for Image-Guided Therapy, NIH P41 EB015898. Dr. Vosburgh also received support from the Center for Integration of Medicine and Innovative Technology (CIMIT).

References

1. GE PET/CT+MR Systems. http://www3.gehealthcare.com/en/Products/Categories/Magnetic_Resonance_Imaging#tabs/tab-46496FE3B3FB45898C8D759AA853A8E3. Accessed August 14, 2013.
2. Siemens Healthcare. http://www.medical.siemens.com/. Accessed August 14, 2013.
3. Vital Images. http://www.vitalimages.com/home.aspx. Accessed August 14, 2013.
4. AMIRA 3D Analysis Software for Life Sciences [FEI Website]. http://www.amira.com/. Accessed August 14, 2013.
5. Mayo Clinic Analyze MD. http://mayoresearch.mayo.edu/mayo/research/robb_lab/analyze-md.cfm. Accessed August 14, 2013.
6. Osirix Imaging Software. http://www.osirix-viewer.com/. Accessed August 14, 2013.
7. 3D Slicer. http://www.slicer.org. Accessed August 14, 2013.
8. Gering D, Nabavi A, Kikinis R, Hata N, O'Donnell LJ, Grimson WEL, Jolesz FA, Wells 3rd WM, Black PM. An integrated visualization system for surgical planning and guidance using image fusion and an open MR. J Mag Res Imaging. 2001;13:967–75.
9. The Visualization Toolkit (VTK). http://www.vtk.org. Accessed August 14, 2013.
10. National Library of Medicine Insight Segmentation and Registration Toolkit (ITK). www.itk.org. Accessed August 14, 2013.
11. NA-MIC Kit. http://wiki.na-mic.org/Wiki/index.php/NA-MIC-Kit. Accessed August 14, 2013.
12. Slicer 4 Guided Tour with MRML Tree. http://www.na-mic.org/Wiki/images/4/42/2011_Summer-Slicer4.ppt. Accessed August 14, 2013.
13. 3D Slicer and Extensible Neuroimaging Archive Toolkit (XNAT). http://wiki.na-mic.org/Wiki/index.php/Projects:ARRA:SlicerWF#Quarterly_Progress. Accessed August 14, 2013.
14. Git. http://git-scm.com/. Accessed August 14, 2013.
15. CMake. http://www.cmake.org/Wiki/CMake. Accessed August 14, 2013.
16. CDash. http://www.cdash.org/cdash/project/about.html. Accessed August 14, 2013.
17. Gering D. Recognizing deviations from normalcy for brain tumor segmentation. PhD thesis, Massachusetts Institute of Technology; 2003.
18. DICOM. http://medical.nema.org/. Accessed August 14, 2013.
19. Simple_Region_Growing-Documentation-3.6 on Slicer. Org. http://www.slicer.org/slicerWiki/index.php/Modules:Simple_Region_Growing-Documentation-3.6. Accessed August 14, 2013.
20. FastMarchingSegmentation-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:FastMarchingSegmentation-Documentation-3.6. Accessed August 14, 2013.
21. EMSegmenter-Tasks on Slicer.org. http://wiki.slicer.org/slicerWiki/index.php/EMSegmenter-Tasks. Accessed August 14, 2013.
22. VMTKEasyLevelSetSegmentation on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:VMTKEasyLevelSetSegmentation. Accessed August 14, 2013.
23. Editor-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:Editor-Documentation-3.6. Accessed August 14, 2013.
24. GrowCutSegmentation-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:GrowCutSegmentation-Documentation-3.6. Accessed August 14, 2013.
25. ModelMaker-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:ModelMaker-Documentation-3.6. Accessed August 14, 2013.
26. DiffusionMRIWelcome-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:DiffusionMRIWelcome-Documentation-3.6. Accessed August 14, 2013.
27. DicomToNRRD-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:DicomToNRRD-3.6. Accessed August 14, 2013.
28. JointRicianLMMSEImageFilter-Documentation-3.6on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:JointRicianLMMSEImageFilter-Documentation-3.6. Accessed August 14, 2013.
29. DiffusionTensorEstimation-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:DiffusionTensorEstimation-Documentation-3.6. Accessed August 14, 2013.

30. Coordinate Systems on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Coordinate_systems. Accessed August 14, 2013.
31. Johnson H, Harris G, Williams K. BRAINSFit: mutual information registrations of whole-brain 3D images, using the insight toolkit in Insight Journal-2008. http://www.insight-journal.org/download/viewpdf/180/8169/bitstream. Accessed August 14, 2013.
32. RegistrationDocumentation:RegLibTable on NA-MIC.org. http://na-mic.org/Wiki/index.php/Projects:RegistrationDocumentation:RegLibTable. Accessed August 14, 2013.
33. VolumeRendering-Documentation-3.6 on Slicer.org. http://www.slicer.org/slicerWiki/index.php/Modules:VolumeRendering-Documentation-3.6. Accessed August 14, 2013.
34. OpenIGTLink. http://openigtlink.org/. Accessed August 14, 2013.
35. Golby AJ, Kindlmann G, Norton I, Yarmarkovich A, Pieper S, Kikinis R. Interactive diffusion tensor tractography visualization for neurosurgical planning. Neurosurgery. 2011;68(2):496–505.
36. Shenton M, et al. A review of magnetic resonance imaging and diffusion tensor imaging findings in mild traumatic brain injury. Brain Imaging Behav. 2012;6(2):137–92.
37. Irimia A, Chambers MC, Alger JR, Filippou M, Prastawa MW, Wang B, Hovda D, Gerig G, Toga AW, Kikinis R, Vespa PM, Van Horn JD. Comparison of acute and chronic traumatic brain injury using semi-automatic multimodal segmentation of MR volumes. J Neurotrauma. 2011;28:2287–306.
38. Pilato CM, Collins-Sussman B, Fitzpatrick BW. Version control with submission, 2nd Edn. O'Reilly Media, Inc, Sebastopol, CA. 2008.
39. http://www.slicer.org/slicerWiki/index.php/Slicer4:Extensions. Accessed November 6, 2013.
40. http://www.slicer.org/slicerWiki/index.php/Slicer_3.6:Training. Accessed November 6, 2013.