

SlicerRT: Radiation therapy research toolkit for 3D Slicer

Csaba Pinter^{a)} and Andras Lasso

Laboratory for Percutaneous Surgery, School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada

An Wang and David Jaffray

Radiation Medicine Program, Princess Margaret Hospital, University Health Network, Toronto, Ontario M5G 2M9, Canada

Gabor Fichtinger

Laboratory for Percutaneous Surgery, School of Computing, Queen's University, Kingston, Ontario K7L 3N6, Canada

(Received 14 June 2012; revised 19 August 2012; accepted for publication 10 September 2012; published 27 September 2012)

Purpose: Interest in adaptive radiation therapy research is constantly growing, but software tools available for researchers are mostly either expensive, closed proprietary applications, or free open-source packages with limited scope, extensibility, reliability, or user support. To address these limitations, we propose SlicerRT, a customizable, free, and open-source radiation therapy research toolkit. SlicerRT aspires to be an open-source toolkit for RT research, providing fast computations, convenient workflows for researchers, and a general image-guided therapy infrastructure to assist clinical translation of experimental therapeutic approaches. It is a medium into which RT researchers can integrate their methods and algorithms, and conduct comparative testing.

Methods: SlicerRT was implemented as an extension for the widely used 3D Slicer medical image visualization and analysis application platform. SlicerRT provides functionality specifically designed for radiation therapy research, in addition to the powerful tools that 3D Slicer offers for visualization, registration, segmentation, and data management. The feature set of SlicerRT was defined through consensus discussions with a large pool of RT researchers, including both radiation oncologists and medical physicists. The development processes used were similar to those of 3D Slicer to ensure software quality. Standardized mechanisms of 3D Slicer were applied for documentation, distribution, and user support. The testing and validation environment was configured to automatically launch a regression test upon each software change and to perform comparison with ground truth results provided by other RT applications.

Results: Modules have been created for importing and loading DICOM-RT data, computing and displaying dose volume histograms, creating accumulated dose volumes, comparing dose volumes, and visualizing isodose lines and surfaces. The effectiveness of using 3D Slicer with the proposed SlicerRT extension for radiation therapy research was demonstrated on multiple use cases.

Conclusions: A new open-source software toolkit has been developed for radiation therapy research. SlicerRT can import treatment plans from various sources into 3D Slicer for visualization, analysis, comparison, and processing. The provided algorithms are extensively tested and they are accessible through a convenient graphical user interface as well as a flexible application programming interface.

© 2012 American Association of Physicists in Medicine. [<http://dx.doi.org/10.1118/1.4754659>]

Key words: radiation therapy, 3D Slicer, DICOM-RT, dose volume histogram, dose comparison

I. INTRODUCTION

The aim of adaptive radiation therapy is to improve therapeutic results by the adaptation of the treatment plan to the patient over the course of the treatment. To this end, the radiation therapy (RT) community has traditionally employed a variety of commercial and in-house built tools for image analysis and treatment planning with a few examples of coordinated software development.

The feature set of the commercial software applications typically cover only routine clinical procedures and, due to the strict regulation of commercial medical devices, their flexibility and extensibility are usually very limited. Considering the variety of software applications and the complexity of RT

treatment plans, reproducing and comparing results produced by different software tools or by different researchers is a demanding and error-prone task.

There have been initiatives for open-source RT research toolkits, such as CERR,¹ PLUNC,² dicompyler,³ or MMCTP.⁴ As none of these toolkits fulfill all the required functions for a powerful RT research platform, their potential to become a standard platform for RT research is limited. Shortcomings include dependency on commercial closed-source software packages (MATLAB), restrictive license (not free, not open-source, not freely usable for commercial purposes, etc.), large and nonmodular code base, lack of documentation, insufficient user and developer support, dependency on proprietary data formats, difficult extensibility, and

limited feature set. At the same time, as a result of extensive and coordinated efforts, a comprehensive open-source software application framework, 3D Slicer⁵ has been developed for medical image computing and image-guided interventions. 3D Slicer provides functions and solutions to many challenges of RT research software platform development.

We propose the SlicerRT toolkit as a common software platform to address the limitations of existing software solutions. By adopting 3D Slicer, RT researchers can gain access to a well-established, widely used software application and multiple state-of-the-art computational algorithm toolkits. 3D Slicer also fulfills the requirements of flexibility, extensibility, free, non-restricted use, full source code, data, and documentation availability, support of common data formats, widely known development environment, ease of use, comprehensive data visualization and numerical analysis of research data, and flexible data import and export.

II. METHODS

Key system requirements, features, and priorities were identified through a series of consensus discussions with RT researchers, and based on clinically relevant use cases. The participating researchers are members of the Ontario Consortium for Adaptive Intervention in Radiation Oncology (OCAIRO), which is a diverse community covering over 20 RT research groups in Ontario, Canada, including Cancer Centre of South Eastern Ontario, London Health Sciences Centre, Queen's University, Robarts Research Institute, Sunnybrook Health Sciences Centre, The Ottawa Hospital Regional Cancer Centre, and University of Toronto.

Many required functionalities, such as visualization of structure sets and anatomical images, multimodality image fusion were already part of the 3D Slicer core feature set. The following additional RT-specific key features were identified: (a) robust DICOM-RT import and export for data exchange with commercial RT planning systems, (b) visualization of structures, dose distributions, and treatment plans, and (c) dose calculation and conversion, accumulation, visualization, and comparison, dose volume histogram (DVH,⁶) and related metrics.

II.A. Platform

The required features were implemented using 3D Slicer as application platform. 3D Slicer provides an immense amount of functionality to visualize and analyze a wide range of datasets, such as anatomical/functional images, image segmentation results, surface models, point sets, and rigid and nonrigid transformations. It contains state-of-the-art algorithms that are also fundamental tools of RT research, such as image segmentation and registration (rigid, affine, and deformable). It supports import and export data from a wide range of standard data formats, including DICOM.

3D Slicer is an end-user application, but it also provides many opportunities for customization and extension. 3D Slicer provides an interactive console that gives access to all the computational algorithms and loaded data objects using

Python scripting language. Similarly to MATLAB, the console allows performing custom calculations on the fly without modifying and building the source code. 3D Slicer also enables batch processing, i.e., running the desired computation on multiple inputs without user interaction. Custom modules can be developed in either Python (for greater flexibility, simpler modification), or C++ (for better performance, more efficient memory management).

3D Slicer is open-source, with a BSD-style⁷ license that allows free, unrestricted use for even commercial purposes. Very importantly, 3D Slicer has been used by a large and steadily growing user community in over 100 different research projects since 2003.⁸ 3D Slicer has been downloaded more than 40 000 times.⁹

II.B. Design

3D Slicer's plugin mechanism enables the rapid development of custom modules in different programming languages and levels of integration. The *command-line interface* modules are standalone executables operating with built-in data types and offering basic user interaction (selection of input and output data and custom parameter values). 3D Slicer automatically generates the user interface based on a simple text file that describes the input and output data types of the algorithm, therefore the developers do not have to spend time with user interface implementation. The *loadable modules* are plugins implemented in the C++ language that are integrated tightly with 3D Slicer. These modules have access to all other modules and the internals of the application and they can define custom, interactive graphical user interfaces. The *scripted modules* are written in the Python language. These modules can be developed and modified without rebuilding or restarting 3D Slicer and they have similar access to the application internals as loadable modules.

The SlicerRT features are implemented as multiple plugin modules, bundled as one downloadable extension. Each module is independent, addressing only a specific requirement. This modularization makes development and maintenance work much easier compared to working with a monolithic code base. All modules in the SlicerRT extension are implemented as loadable modules, because this allows

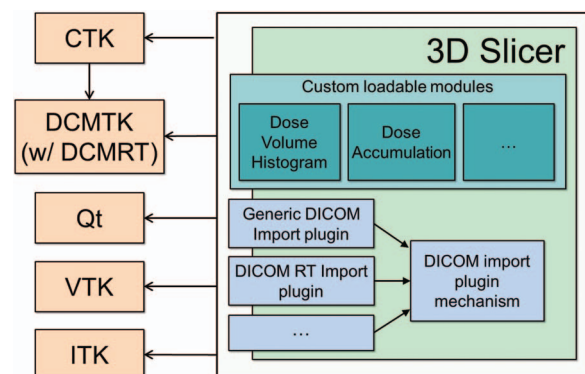


FIG. 1. Architecture of SlicerRT: Custom loadable modules added in 3D Slicer.

for a nuanced customization of the user interface and for the optimization of computation performance.

The architecture of SlicerRT is shown in Fig. 1. SlicerRT uses the CTK (Ref. 10) and Qt (Ref. 11) packages for user interface widgets and database handling, the VTK (Ref. 12) library for visualization and mesh processing, and the ITK (Ref. 13) library for image processing and registration. For DICOM support, it uses the DCMTK (Ref. 14) software library with a recently released module, DCMRT, that supports reading and writing of radiation therapy information objects.¹⁵

All objects (volumetric images, surface models, transforms, etc.) in 3D Slicer are stored in a hierarchical structure of nodes encoded in the medical reality modeling language (MRML). Each MRML node has its own list of custom attributes that can be used to specify additional characteristics for the data object. This method of storage enables the modules to have access to the MRML tree, allowing new extensions to leverage existing processing and visualization functions without directly interfering with other modules.

SlicerRT extends the functionality of the 3D Slicer DICOM reader with a custom plugin to create customized MRML nodes from DICOM RT information objects:

- RT structure set: Point structures are loaded as a series of points (into *vtkMRMLAnnotationFiducialNode* objects). Contours are loaded as a series of “ribbons,” i.e., the structure contours expanded in 3D as band surfaces (into *vtkMRMLModelNode* objects). The imported contours and points are organized under a hierarchy node parent.
- RT dose: The dose volume is loaded as a volumetric image (into a *vtkMRMLVolumeNode* object). The properties of the dose volume, such as the dose unit and the dose unit scaling are stored as custom attributes.
- RT plan: In the current version, the isocenter points are loaded (as *vtkMRMLAnnotationFiducialNode* objects).
- RT image: Loaded as volumetric image slices (into *vtkMRMLVolumeNode* objects).

All the data generated by SlicerRT are also stored in MRML nodes. The computed DVH curves are stored in scalar arrays (*vtkMRMLDoubleArrayNode* objects) and displayed using chart nodes (*vtkMRMLChartNode* objects). The intermediate binary volume used in DVH computation is a standard label map image (*vtkMRMLVolumeNode* object).

II.C. Development processes

The source code and test data used for building and testing SlicerRT modules are stored in an openly accessible, centralized revision control system. The revision control system is integrated with an issue tracking system that keeps record of all bugs, enhancements, and other tasks. The project has a wiki page, which contains detailed guides for users and developers, as well as information about events and meetings. The project also has a messaging system where users and contributors can ask questions from developers. The Assembla service,¹⁶ a collaborative development management tool, provides the requisite services, through a web-based interface.

II.D. Implementation

The SlicerRT modules are written in the C++, the same programming language as the majority of the 3D Slicer core. The natively compiled C++ code typically runs faster than those implemented in interpreted, scripting programming languages. The native code also enables more control over memory management and easier debugging. For the same reasons, SlicerRT modules can also be executed as standalone programs, while toolkits that require run-time environments are usually launched from within their environment. Inheriting from 3D Slicer, SlicerRT is available for Windows, Linux, and Mac OS X platforms.

The *DICOM-RT import module* is a plugin for 3D Slicer’s DICOM management module. When the user selects a patient in the user interface of 3D Slicer, the import mechanism allows each DICOM import module to interpret the data. The DICOM import modules parse the data and return a list of items that they can read from the selected DICOM objects. The DICOM management module displays the consolidated list of all loadable objects to the user, and loads the items that are selected by the user.

The *dose volume histogram module* (shown in Fig. 2) enables the user to compute and display the DVH curve from a dose volume and structure set. DVH metrics, such as volume, minimum, maximum, and mean dose, V_{dose} (the volume that is radiated by a dose greater than a specified dose), and D_{volume} (the minimum dose in the most exposed portion of a certain volume in a structure) can be also computed.

The *dose accumulation module* computes accumulated dose for multiple input dose volumes as a weighted sum of the dose values. The weight of each input volume is specified in the module’s user interface. The dose volumes have to be spatially aligned prior the dose accumulation to obtain a meaningful summation. This can be achieved by utilizing image registration algorithms already available in the 3D Slicer core. First, the alignment transformation is computed by registering the anatomical volumes and then the same transformation is applied to the corresponding dose volumes.

The *dose comparison module* computes the difference between two coregistered dose volumes using the gamma dose distribution comparison method.^{17,18} This module uses the Plastmatch¹⁹ RT software package internally.

The *isodose module* creates isosurfaces from an input dose volume and a series of defined isodose levels. The output is a series of surface models grouped under a hierarchy node parent that can be visualized either in the 3D viewer or overlaid on image slices.

II.E. Verification and validation

The verification of SlicerRT is an automated process that is driven by scheduled tasks that run the automated testing process on different hardware and software configurations on a regular basis, typically every night. If any problem is detected on any operating system, then the responsible developer receives automatic notification of the issue. This mechanism allows running full regression testing, and ensures

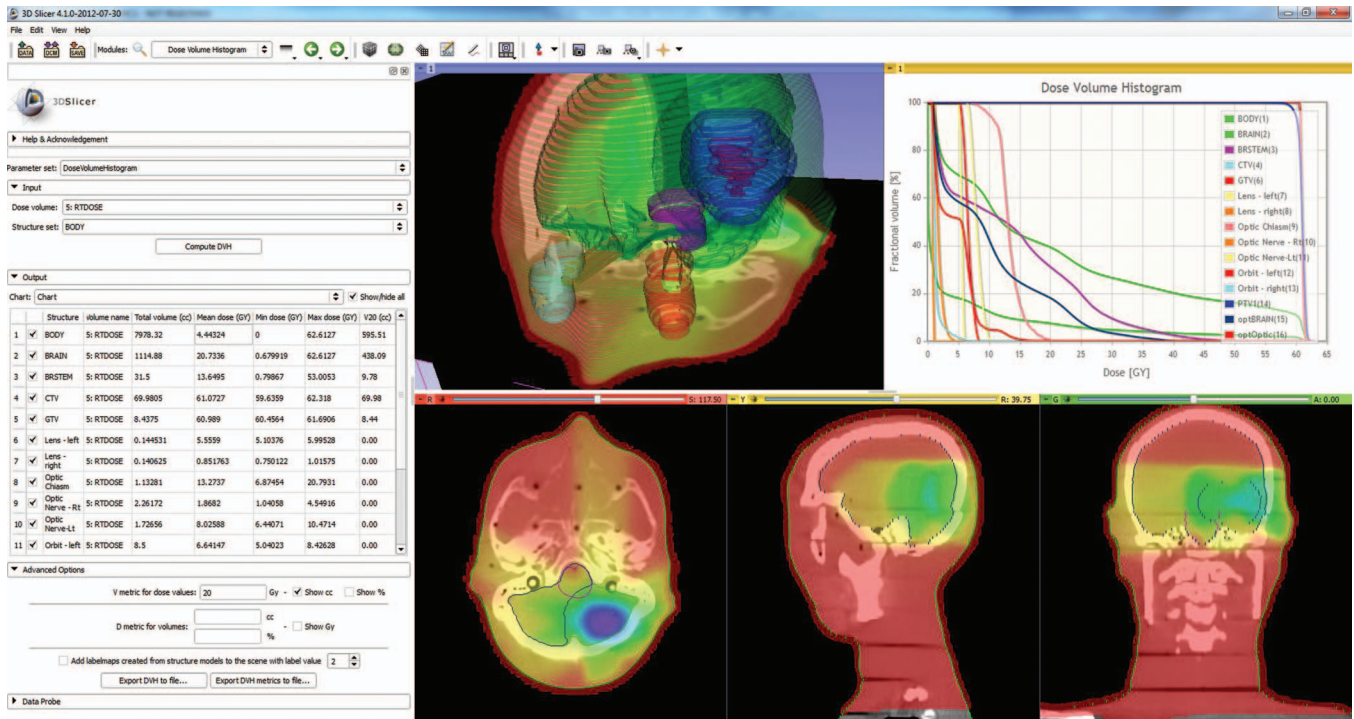


FIG. 2. Dose volume histogram module in a standard 3D Slicer layout with basic visualization. The module toolbox can be seen on the left side, and the DVH chart is found in the upper right corner.

consistent software quality by allowing detection and fixing of problems quickly. In the course of a nightly test, the test cases run one by one, driven by the CTest²⁰ test system. Each case represents a test application with representative input data and baseline to which the computed result is compared. Test results are aggregated on CDash,²¹ a web-based dashboard system (shown in Fig. 3) that provides graph views, summary statistics, and detailed listing of run-time events (warnings, errors, etc.) for each collaborating development site.

The same infrastructure is used to validate the algorithms used in SlicerRT by comparing the outputs to reliable independent computation results. In this case, the script compares the baseline data acquired from another system with the SlicerRT output using a similarity measure. Figure 4 illustrates the data flow of the automatic verification and validation process.

To validate the dose volume histogram module in SlicerRT the computed results were compared to the output of a com-

mercial treatment planning software (Eclipse: EclipseTM radiation therapy treatment planning system, Varian Medical Systems, Inc.) and a research application (CERR). Anatomical images were acquired with computed tomography (CT). The images were loaded into Eclipse, structure sets were contoured, and treatment plans were created. DVH curves were computed in Eclipse and the computed dose map and structure sets were exported to DICOM files. The DICOM data were imported into CERR and DVH curves were computed. The DVH curve values computed by Eclipse and CERR served as validation baseline data.

The DVH curve comparison method described in Ref. 22 was used for comparing the baseline to the results computed by SlicerRT. The comparison method is similar to the gamma analysis used for spatial comparison of dose distributions. A volume-difference criterion of ΔV (as a percentage of the total structure volume) and dose-to-agreement criterion of ΔD (as a percentage of the maximum dose) are specified as an

Extensions-nightly										
Site	Build name	Update	Configure		Build		Test			Build time
		Files	Error	Warn	Error	Warn	Not run	Fail	Pass	
factory.kitwarein.com	20725-SlicerRT-svn212-g++-64bits-Qt4.7-Release		0	0	0	50	0	0	13	9 h ago
factory.kitwarein.com	20725-SuperBuildLoadableExtensionTemplate-svn20725-g++-64bits-Qt4.7-Release		0	0	0	20	0	0	2	9 h ago
factory-ubuntu	20725-SlicerRT-svn212-g++-64bits-Qt4.7-Release		0	0	0	19	0	0	13	11 h ago

FIG. 3. Part of the dashboard showing the results of the SlicerRT automatic tests.

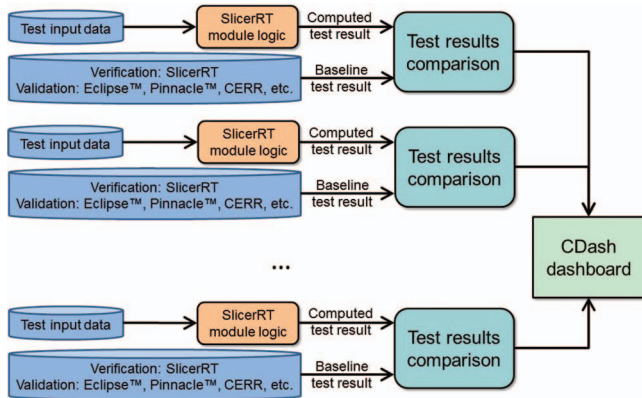


FIG. 4. Data flow diagram of the automatic verification and validation process. The logic class of a particular SlicerRT module being tested computes the output using the input specified for the test. For validation, the data exported by other RT software applications (e.g., Pinnacle™ radiation therapy planning system, Koninklijke Philips Electronics N.V., Eclipse, CERR) is used as baseline. For verification, previous computations by SlicerRT is used as baseline. The baseline data are compared with the current SlicerRT output and the results are submitted to the dashboard.

input to the metric, which are used as normalization factors for the volume and dose differences. The gamma value at a specific histogram point is computed as the square root of the sum of squared normalized differences of the dose and the volume. Therefore, a gamma value of 1.0 corresponds to a volume difference of ΔV and dose difference of ΔD between the two histograms at that point.

III. RESULTS AND DISCUSSION

III.A. Validation results

Anatomical images of pelvis, head, and chest Rando® phantoms were used as test data. Altogether 22 structure sets were contoured and three treatment plans were created. Numerical results of validation against CERR and Eclipse are shown in Table I.

TABLE I. Percentage of structure sets where gamma value was smaller than 1.0 in 95% of the DVH points computed by SlicerRT and the compared software application. ΔD and ΔV were defined as percentage of maximum dose and total structure volume, respectively.

(A) Comparison against CERR				
ΔD ΔV	1%	2%	5%	10%
1%	86%	86%	95%	100%
2%	86%	86%	95%	100%
5%	86%	86%	95%	100%
10%	91%	91%	95%	100%
(B) Comparison against Eclipse				
1%	86%	91%	91%	95%
2%	86%	91%	95%	95%
5%	91%	95%	95%	100%
10%	91%	95%	95%	100%

TABLE II. Speed measurement results comparing CERR and SlicerRT.

Process	CERR (s)	SlicerRT (s)
DICOM import (parsing the files from local storage and building the patient database)	7.4	1.1
Load study and convert the DICOM data into the internal formats	30.0	15.5
DVH tables computation	2.8	1.7 ^a

^aThe sampling of the DVH values in SlicerRT are ten times denser than in CERR.

III.B. Speed comparison

We performed time measurements for SlicerRT and CERR, to demonstrate the performance advantages of the natively compiled over the interpreted code (see Table II). The measurements were performed on a computer with an Intel Core2 Quad CPU (3.4 GHz) and 16 GB of system memory, running 64-bit Microsoft Windows 7. Each measurement was repeated six times and the averages of these are reported. This SlicerRT implementation used algorithms available in the 3D Slicer core libraries in a straightforward manner without performance optimization.

III.C. Example use cases

In this section, use cases are presented to demonstrate the capabilities of 3D Slicer with the SlicerRT extension for addressing clinically relevant radiation therapy research problems.

III.C.1. Use Case 1: Evaluate the effectiveness of RT plan adaptation

It has been shown that patient motion and tumor shrinkage during the course of radiotherapy could have an impact on the quality of a pulsed dose rate brachytherapy plan. Adaptation of plans has been proposed to increase the target coverage and reduce the complication risk of organs at risk (OAR). To examine the effectiveness of plan adaptation, actual delivered doses need to be calculated to verify proper dose distribution on the target. In this use case, we demonstrate dose accumulation of an actual cervical cancer patient using SlicerRT for dose verification after adaptive radiation therapy (ART).

This patient underwent planning and two daily pelvic MRI scans during the treatment. The coverage of clinical target volume (CTV) was not acceptable with the original plan at the day 2 MRI. Online plan adaptation was employed using the two daily MR images to satisfy clinical requirements. We then compared the delivered dose between the original plan and the adapted plan using dose accumulation. To accumulate the delivered dose, deformable registration was employed using the B-spline option in the *General Registration (BRAINS)* module to align daily MR images to day 1 planning MR image. Next, daily dose distribution maps were warped using the *Resample Image (BRAINS)* module with the resulting non-rigid transformation. They were then added to generate the final accumulated dose using SlicerRT's dose accumulation module (shown in Fig. 5). DVH for CTV using accumulated

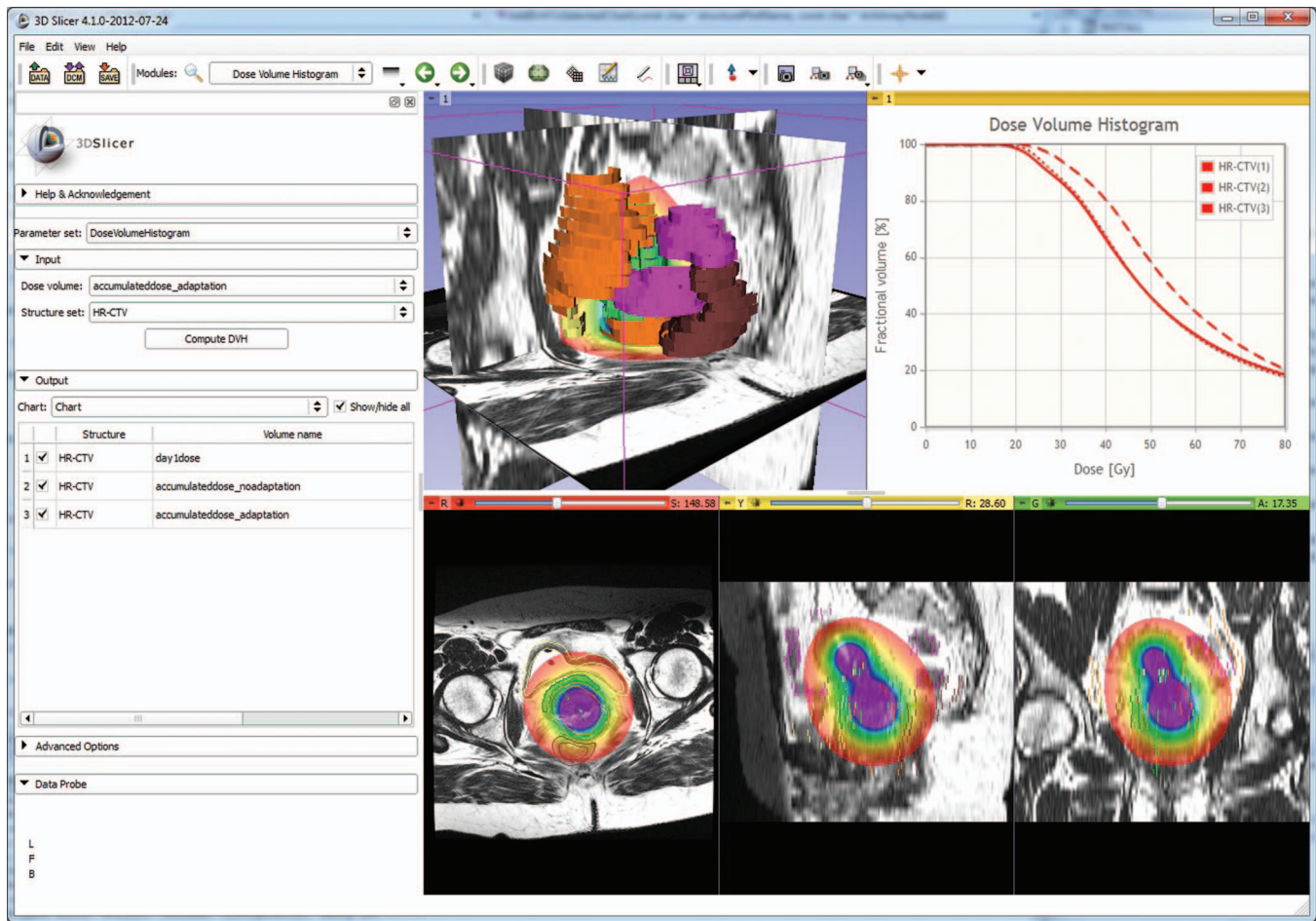


FIG. 5. Dose accumulation module in a standard 3D Slicer layout. The fused anatomical and dose volumes can be seen on the bottom, the structures together with the anatomy in the 3D view in the upper middle area, and the compared DVH charts using different line styles in the upper right corner.

dose matrices were then generated using SlicerRT's dose volume histogram module and D_{90} metrics were then compared.

D_{90} from the original plan was reduced to 28.5 Gy from the prescribed dose at 36 Gy due to risk of overdose to OAR. The accumulated D_{90} without plan adaptation and with plan adaptation was 27.3 Gy and 33.5 Gy, respectively. The D_{90} from the adapted plan was much closer to the prescribe dose since doses in day 2 and day 3 were elevated without increase of dose to OAR. The impact of plan adaptation can be evaluated using the deformable registration and dose accumulation module of SlicerRT.

III.C.2. Use Case 2: Evaluation of improvements resulting from isocenter shifting

One of the ART techniques is isocenter shifting,^{23–25} in course of which the planned dose distribution is translated so that the target volume is fully covered and the least possible margin can be used. The shift can be determined automatically by using the image-based registration methods available in 3D Slicer, using the pretreatment image as the reference volume, and the planning scan as the moving volume. Alternatively, the shift can be manually set by utilizing 3D Slicer's image fusion capabilities while adjusting the param-

eters of a rigid transformation applied on the planning image. To quantify the improvements made on the treatment plan, DVH is computed using the *dose volume histogram* module, using both the original, and the transformed dose distributions with the structures contoured on the pretreatment plan as the inputs. A specific D metric (e.g., D_{95}) can then be used to compare the original and transformed dose distributions to evaluate the improvement from shifting the isocenter.

III.C.3. Use Case 3: Implementation of a custom isocenter shift method

A variant of the isocenter shift method²⁶ determines the translation by computing the center-of-mass of the target soft tissue volume in both the planning image and the daily scan. The beams then can be realigned by the vector difference of these center-of-mass coordinates. A short Python script can be developed to calculate the center-of-mass points. All the input data for the computation are available for the script in the MRML nodes, the script can be edited and executed while 3D Slicer is running. The development does not require any additional software installation or development environment setup, as the script and commands can be either entered into the Python console window, or loaded from a simple text file.

III.D. Licensing and collaboration strategy

3D Slicer and SlicerRT are distributed under a BSD-style open-source license that contains no restrictions on the use of the software. The software may be used not only for research purposes but also in clinical and commercial projects. Note, however, that validation for a particular clinical purpose is an onus of the user. To leverage the efforts of the SlicerRT user and developer community, any custom enhancements or fixes are encouraged to be contributed back to the common platform.

III.E. Future work

Further development of SlicerRT includes extension of the validation and the functionalities of existing RT modules and creating new modules for further improving RT specific data visualization, processing and export capabilities. Integration with Plastimatch provides access to a number of other RT-related computational algorithms that will be leveraged in new SlicerRT modules. SlicerRT can be extended to utilize algorithms from other software packages as well. Through the enhancement of the 3D Slicer platform it is expected that SlicerRT modules will be able to use functions of MATLAB and the Insight Toolkit (ITK,¹³). The issue tracking system on the project website provides up-to-date information about on-going and future developments.

III.F. Distribution

SlicerRT modules can be downloaded as an extension for 3D Slicer 4.1 or higher. Source code and build instructions are available on the project website. Current and prospective users are encouraged to visit the project website¹⁶ and to share their comments, ideas, and needs.

III.G. Summary

SlicerRT aims to become a complete toolkit for RT researchers by providing most frequently used features for objective comparison and analysis of RT data and allowing researchers to easily develop and share their additional custom methods. Current funding ensures continuous development for several years and we expect that other research groups will also join these efforts.

ACKNOWLEDGMENTS

SlicerRT has been made possible by funding from the Province of Ontario, Ontario Research Fund Research Excellence Program, Ontario Consortium for Adaptive Interventions in Radiation Oncology (OCAIRO), and An Applied Cancer Research Unit of Cancer Care Ontario with funds provided by the Ministry of Health and Long-Term Care. G.F. was funded as a Cancer Care Ontario Research Chair. Test data were provided by John Schreiner, Johnson Darko, and Greg Salomons (Kingston General Hospital, Kingston, ON,

Canada) and by Young-Bin Cho and Seungjong Oh (Radiation Medicine Program, Princess Margaret Hospital, University Health Network, Toronto, ON, Canada).

^{a)} Author to whom correspondence should be addressed. Electronic mail: pinter@cs.queensu.ca

¹ J. O. Deasy, A. I. Blanco, and V. H. Clark, "CERR: A computational environment for radiotherapy research," *Med. Phys.* **30**(5), 979–985 (2003).

² <http://planunc.radonc.unc.edu>.

³ <http://code.google.com/p/dicompyler>.

⁴ A. Alexander, F. DeBlois, G. Stroian, K. Al-Yahya, E. Heath, and J. Seintjens, "MMCTP: A radiotherapy research environment for Monte Carlo and patient-specific treatment planning," *Phys. Med. Biol.* **52**, N297–N308 (2007).

⁵ S. Pieper, M. Halle, and R. Kikinis, 3D SLICER. *Proceedings of the 1st IEEE International Symposium on Biomedical Imaging: From Nano to Macro* (Brigham and Women's Hospital, Boston, MA, 2004), pp. 632–635. The paper can be found (and citations exported) at <http://www.slicer.org/publications/item/view/91>.

⁶ R. E. Drzymala, R. Mohan, L. Brewster, J. Chu, M. Goitein, W. Harms, and M. Urie, "Dose-volume histograms," *Int. J. Radiat. Oncol., Biol., Phys.* **21**(1), 71–78 (1991).

⁷ <http://opensource.org/licenses/BSD-2-Clause>.

⁸ http://www.slicer.org/pages/Slicer_Community.

⁹ <http://download.slicer.org/stats>.

¹⁰ <http://www.commontk.org>.

¹¹ <http://qt.nokia.com>.

¹² W. Schroeder, K. Martin, and B. Lorensen, *Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*, 4th ed. (Kitware, Clifton Park, NY, 2006).

¹³ L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*. (Kitware Inc., Clifton Park, NY, 2003).

¹⁴ M. Eichelberg, J. Riesmeier, T. Wilkens, A. J. Hewett, A. Barth, and P. Jensch, "Ten years of medical imaging standardization and prototypical implementation: The DICOM standard and the OFFIS DICOM Toolkit (DCMTK)," *Proc. SPIE* **5371**, 57–68 (2004).

¹⁵ M. Y. Law and B. Liu, "Informatics in radiology: DICOM-RT and its utilization in radiation therapy," *Radiographics* **29**(3), 655–667 (2009).

¹⁶ <http://www.assembla.com/spaces/slicerrt>.

¹⁷ D. A. Low, W. B. Harms, S. Mutic, and J. A. Purdy, "A technique for the quantitative evaluation of dose distributions," *Med. Phys.* **25**(5), 656–656 (1998).

¹⁸ D. A. Low and J. F. Dempsey, "Evaluation of the gamma dose distribution comparison method," *Med. Phys.* **30**(9), 2455–2464 (2003).

¹⁹ G. C. Sharp, R. Li, J. Wolfgang, G. Chen, M. Peroni, M. F. Spadea, S. Mori, J. Zhang, J. Shackelford, and N. Kandasamy, "Plastimatch: An open source software suite for radiotherapy image processing," in *Proceedings of the XVIIth International Conference on the Use of Computers in Radiotherapy (ICCR)* (Amsterdam, the Netherlands, 2010).

²⁰ http://www.cmake.org/Wiki/CTest/Testing_With_CTest.

²¹ <http://www.cdash.org>.

²² M. A. Ebert, A. Haworth, R. Kearvell, B. Hooton, B. Hug, N. A. Spry, S. A. Bydder, and D. J. Joseph, "Comparison of DVH data from multiple radiotherapy treatment planning systems," *Phys. Med. Biol.* **55**, N337–N346 (2010).

²³ A. B. Hwang, J. Chen, T. B. Nguyen, A. G. Gottschalk, M. R. Roach III, and J. Pouliot, "Irradiation of the prostate and pelvic lymph nodes with an adaptive algorithm," *Med. Phys.* **39**(2), 1119–1124 (2012).

²⁴ D. Thongphiew, Q. J. Wu, W. R. Lee, V. Chankong, S. Yoo, R. McMahon, and F. F. Yin, "Comparison of online IGRT techniques for prostate IMRT treatment: Adaptive vs repositioning correction," *Med. Phys.* **36**(5), 1651–1662 (2009).

²⁵ V. Murthy, Z. Master, P. Adurkar, I. Mallick, U. Mahantshetty, G. Bakshi, H. Tongaonkar, and S. Shrivastava, "'Plan of the day' adaptive radiotherapy for bladder cancer using helical tomotherapy," *Radiother. Oncol.* **99**, 55–60 (2011).

²⁶ W. Song, B. Schaly, G. Baumann, J. Battista, and J. Van Dyk, "Image-guided adaptive radiation therapy (IGART): Radiobiological and dose escalation considerations for localized carcinoma of the prostate," *Med. Phys.* **32**(7), 2193–2203 (2005).