Secondary development based on 3D Slicer extension modules

Liyuan Ma[1], Chaoqi Zhang[2], Weina Wu[3], Jincai Chang[4], Jianzhong Cui[5]

[1, 2, 3, 4]College of Sciences, North China University of Science and Technology, Tangshan, 063210, China
[4]3D Modeling and Application Innovation Laboratory, North China University of Science and Technology, Tangshan, 063210, China
[5]Department of Neurosurgery, Tangshan Gongren Hospital, Tangshan, 063000, China
[4, 5]Corresponding author
E-mail: [1]*15530507027@163.com*, [2]*Jacksonyee0817@163.com*, [3]*15226590131@163.com*, [4]*jincai@ncst.edu.cn*, [5]*zhcui0001@163.com*

Check for updates

Abstract. Slicer module is an important part of software, which provides algorithmic support for data processing for the realization of various functions of software. As an external plug-in that needs to be installed, extensions have strong independence. Developers can redevelop the original extension module and modify its functions and interfaces in order to achieve better results to meet the requirements of medical use. In this paper, the feasibility scheme and specific operation steps for the secondary development of the extended module are proposed, so as to improve efficiency, simplify the operation process, and facilitate the use of Slicer software in clinical medicine and medical research.

Keywords: extension modules, secondary development, compilation, packaging, medical use.

1. Introduction

As an open source software [1] for medical image processing and 3D visualization [2], 3D Slicer has many applications [3] in medicine and research. Slicer platform is based on open source software development such as VTK, ITK, CTK, CMake, Qt and Python. The main programming languages are C++, Python and so on. Slicer architecture [4] can be divided into three layers: basic layer, algorithm layer and application layer. The structure diagram is as shown in Fig. 1.
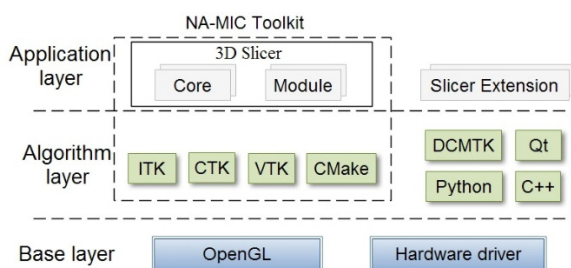

Fig. 1. Slicer architecture diagram

The base layer is provided by the operating system with the lowest level library, such as OpenGL, hardware driver, etc. OpenGL is based on 3D, professional applications, providing high-quality 3D graphics and image performance, to efficiently use the windows and graphics resources provided by the host system, while hardware driver meets accelerated rendering. As the middle layer of the architecture, the algorithm layer is a portable dependency library, including the basic programming language library and some algorithm libraries for image processing. It provides a higher level of functions and abstractions, including CMake, ITK, CTK, VTK and so on. The application layer is Slicer's application, including core, module and extension. The application core implements the user interface, provides the data input/output interface, uses the

algorithm to realize visualization, and reserves the interface for developing new modules and extensions. Modules and extensions implement data processing algorithms, such as segmentation, registration, image filtering and so on.

Secondary development of extension module and modification of its specific algorithm and interface can optimize software functions and beautify its appearance, so as to achieve better results. Chen studied on 3D Slicer's intraoperative implant implantation surgical navigation system [5], developed and verified the SNS module to support the surgical navigation, reliable calibration was carried out on the drilling operation. Point-based positioning and surface matching positioning, real-time tracking and display of the position and orientation of the surgical drill on a computer screen has been achieved, which can reduce the risk of surgery. Saleh Choueib and others added VR function to the open source medical image analysis and visualization platform 3D Slicer, developed 3D Slicer's VR extension-SlicerVirtualReality (SlicerVR) [6], making it a medical virtual reality visualization platform. Wenye Zeng proposed an internationalization method [7] suitable for Slicer software system by studying Qt internationalization system and Slicer software platform interface structure and interface creation method. By developing the extension, the Slicer interface is optimized for medical image processing. In this paper, the feasibility scheme of the secondary development of Slicer extension and specific operational steps are described in detail to assist Slicer software extension functions and beautify the appearance so that the software can be better applied in medicine and research.

2. Basic introduction to Slicer modules and extensions

3D Slicer software relies on extensions and modules to achieve various functions. Among them, the built-in module helps software realize basic functions such as modeling and image processing, and extension is the "package" that delivers new functions to users through extension manager. An extension can contain one or more modules, which are presented to the user as built-in modules after the extension is installed.
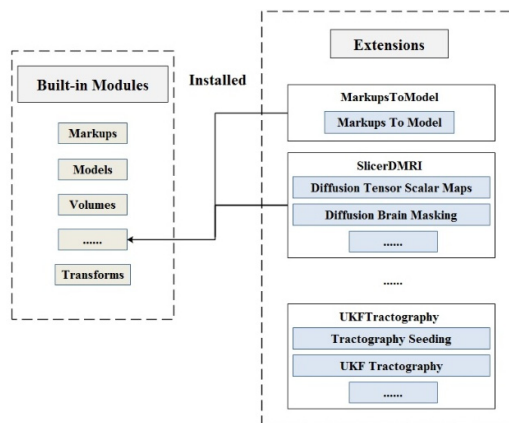

Fig. 2. The diagram of Slicer built-in module and expansion module

There are three types of Slicer modules: Command Line Interface (CLI), Scripted Modules, Loadable Modules. A module may be just one or two or three of these types, but the type of module does not affect the final look and feel of the software. CLI can only communicate through the predefined interface, cannot directly access the core of Slicer, and cannot directly operate the user interface, so it has limited interaction. Script modules are written in Python, more suitable for development, can access most of the internal content of the software, in the case of no DCMTK, ITK, can only access the content encapsulated in Python. Loadable modules are C++ plug-ins built for Slicer that define custom GUI for specific behavior because they have complete control over the application.

Each extension of Slicer has a unique algorithm to achieve corresponding functions. For example, MarkupsToModel has the ability to create a tube or closed surface model from the points marked by the reference node; SlicerDMRI is an umbrella extension of the Slicer Diffusion MRI function; UKF Tractography traces fibers in a DWI Volume using the multiple tensor unscented Kalman Filter methodology.

3. Tools needed for secondary development of extension modules

Before modifying the extension module, we need to build an appropriate environment for it. The tools we need to download include CMake, SVN, Git, Visual Studio, Qt, etc., which are explained respectively [8].

CMake mainly solves cross-platform compilation problems. Both Slicer itself and its extension modules use multiple programming languages, and the compilation process requires multiple compilers. CMake is to use a simple statement to describe the construction process of all platforms, generate a Makefile or Project file, program compilation information is written into the CmakeLists.txt file.

SVN, or Subversion, places the data in an archive repository and records every change to the file, making changes to the code obvious and easy to restore to a previous version. Slicer source code and most of its method libraries are using SVN code management.

Git is an open source distributed version control system. A large part of the libraries used in compiling Slicer also use Git for version control. Slicer does not build its own Git server, and the source code is hosted on GitHub's server.

Visual Studio is an application development program based on Windows platform. It can be used to create Windows applications and network applications. It can also be used to create network services, smart device applications and Office plug-ins. The software is widely used in the compilation, debugging and packaging of source code.

Qt is a cross-platform C++ graphical user interface application development framework that is easy to extend and allows component programming. You can compile the extension module either with the full Qt version compiled or by yourself. This software mainly aims at the modification of module interface in the development process.

The versions of the software required above are related to the version number of Slicer, and different versions of Slicer require different software versions to download.

4. Extension module for secondary development steps

4.1. Download of extension module source code

In order to redevelop the module, the first step is to obtain the source code of the extension module. The source code for the extension module can be downloaded from GitHub or SVN. The following sections describe the download from GitHub, using the MarkupsToModel extension as an example. First, open 3D Slicer to find the Extensions Manager and the module to be downloaded from Install Extensions. Click on its details as shown in Fig. 3(a), right-click View source code and select Code Link URL to copy the download link; or left-click View source code and then click Clone or download to copy the download link, as shown in Fig. 3(b).

After copying the link, go to Git Bash and type "git clone (the folder name of the module source code)" to download it. The download path is user /(user name)/MarkupsToModel "git clone https://github.com/SlicerIGT/SlicerMarkupsToModel MarkupsToModel".

4.2. Compilation and generation of extension module source code

Because the source code is written in many languages, CMake cross-platform software is needed to solve the compilation problem [9]. Open the software, as shown in Fig. 4, select the

storage location of good extension module source code, and then choose the location of the generated file, attention to the path that don't exist in Chinese characters, check the Advanced, add 5 generated path, click the Configure, in the pop-up dialog box, select the generator at the specified project the downloaded version of Visual Studio [10], choose suitable for generator digits, and then click Finish to start compilation. After compiling, click Generate and open Visual Studio to see the code.
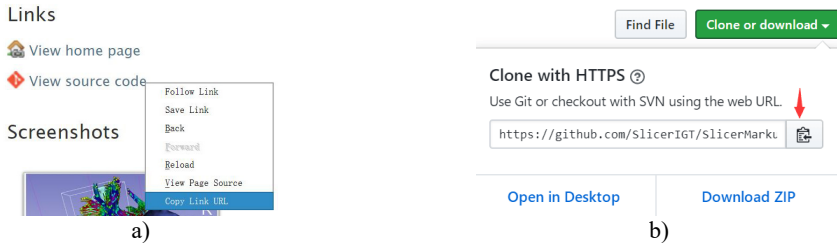
Fig. 3. a) View source location from module details, b) the link of download module source code
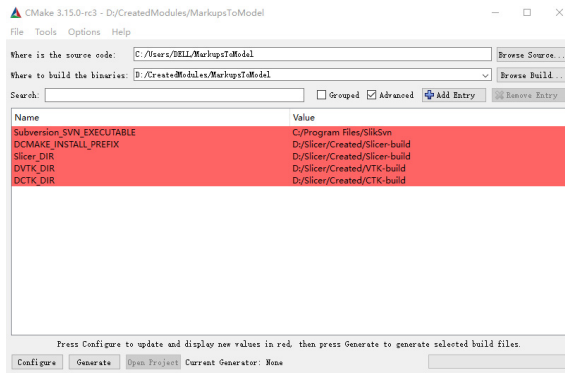

Fig. 4. CMake compilation example

4.3. Packaging of extension modules

Open Visual Studio, right-click ALL-BUILD to generate and debug. After debug success, open Visual Studio, two scenarios may occur after generation:
i) Package project exists;
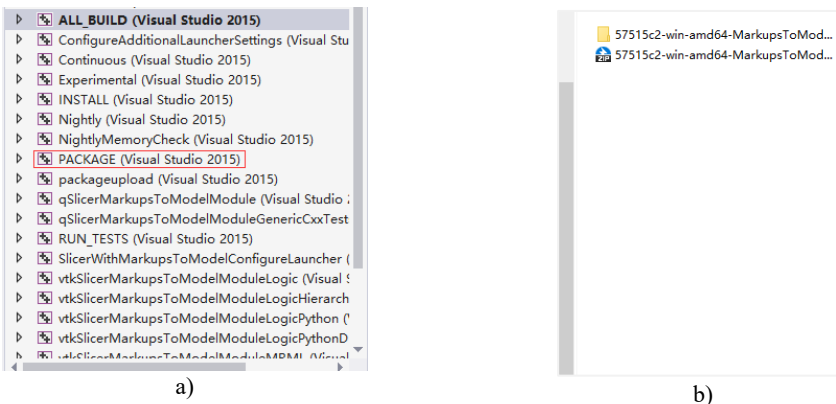ii) No Package project exists.

Fig. 5. a) The program files of MarkupsToModel extension,
b) the package file of MarkupsToModel extension

For the first case, the packaged file can be generated directly. Take the MarkupsToModel extension as an example. The project files are shown in Fig. 5(a). Right-click Package project to generate. After make a success, the _CPack_Packages folder will appear in the extended generation folder. In this folder, there is a package file with the corresponding version number, as shown in Fig. 5(b).

In the second case, the packaged file cannot be generated directly. Take the SlicerDMRI extension as an example, and the project file is shown in Fig. 6(a). Because there is no Package project, you need to right-click the inner project to generate the required files for packaging.
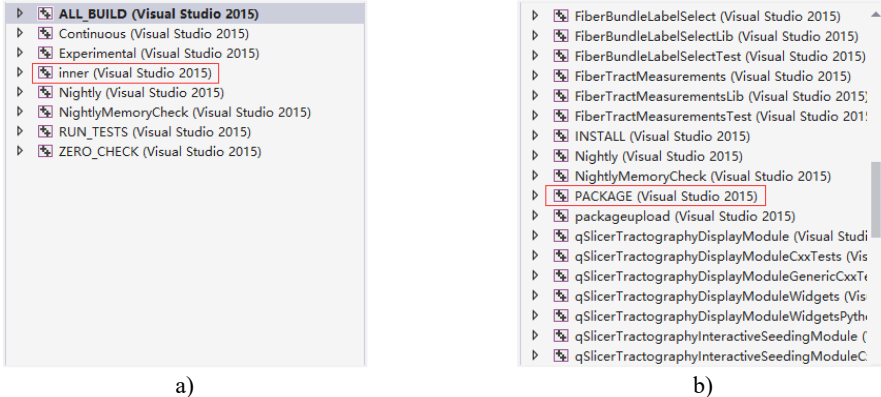

a)   b)

Fig. 6. a) The program files of SlicerDMRI extension, b) the generated package program file

Open the inner-build folder in the extended build folder, as shown in Fig. 7(a), and open the SlicerDMRI.sln file in it. As shown in Fig. 6(b), the project file contains the Package project, right-click generation, after success, the _CPack_Packages folder will appear under the inner-build folder, which has a package file corresponding to the software version number, as shown in Fig. 7(b).
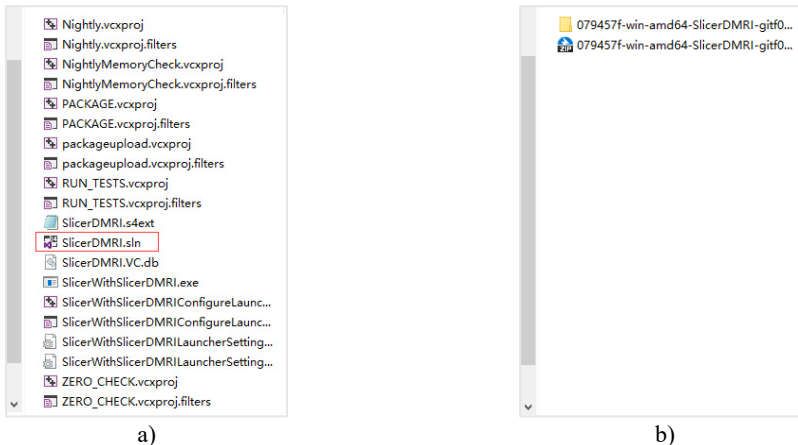

a)   b)

Fig. 7. a) SlicerDMRI.sln file in the inner-build folder, b) the package file of SlicerDMRI extension

4.4. Import of extension module

After the above operation is completed, import the module. When Slicer is opened, there are two methods for module import, which are described respectively below.

Method 1: Go to Edit → Application Settings → Modules → and select package files, as shown in Fig. 9. After a reboot, you will have the extension and its internal modules.
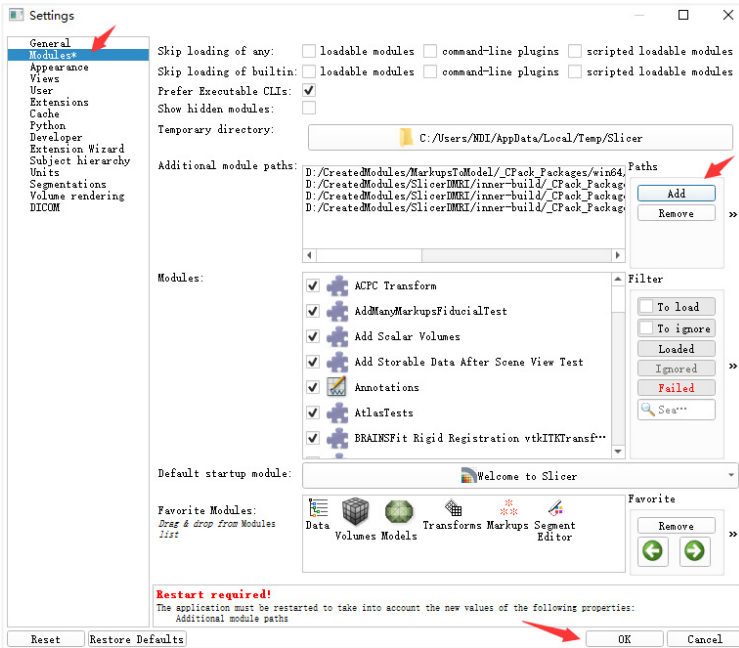
Fig. 8. Module import instance

Fig. 9. Sample of the packaged files to import

Method 2: Open Extensions Manger → Install Extensions → Install Extension from File... → Select the packaged file compression package. You will have the extension and its modules after a reboot.
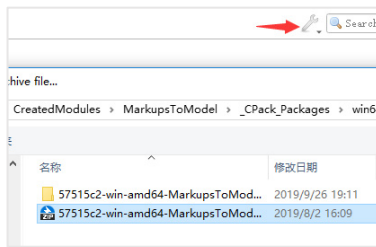

Fig. 10. Import extension module via extension manger

Finally, open All Modules, as shown below, which includes the MarkupsToModel module in the MarkupsToModel extension and the Diffusion Tensor Scalar Maps module in the Slicer DMRI extension, indicating that both extensions were imported, verifying the feasibility of this operation.

5. Conclusions

Because 3D Slicer software provides free and powerful cross-platform [11] processing tools, it is widely used in many fields [12]. On this basis, the method of this article is used for secondary development of its extension module to enrich and improve its functions, beautify its appearance, and enhance the user experience, making the software more widely used in medical care.
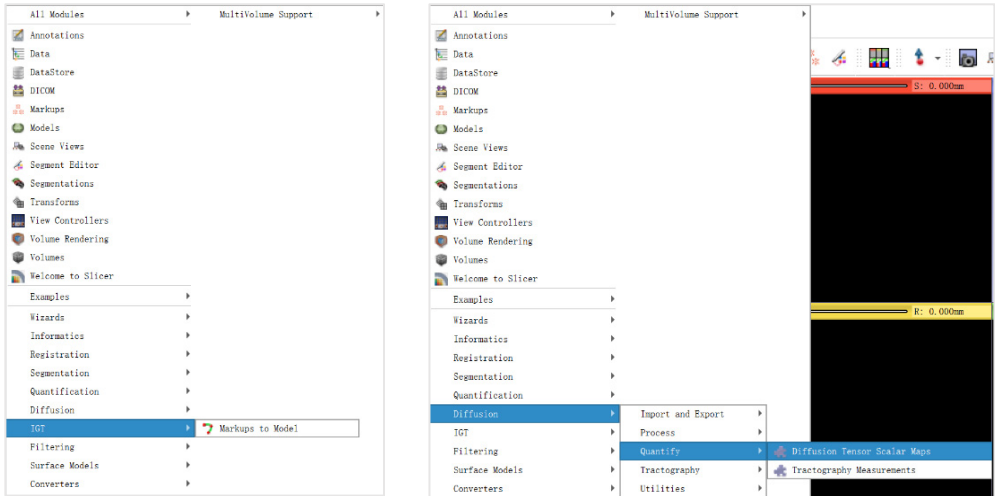
Fig. 11. Modules in both extensions are imported

Especially for meeting the requirements of medical equipment software with high accuracy, high speed, and safety, it is of great significance to optimize the algorithm through secondary development, reduce the software running time and improve the calculation accuracy. For example, by carrying out the secondary development of some of the software's expansion modules, and then modifying and simplifying the software interface, the packaged software is applied to surgical navigation, which is conducive to improving surgical efficiency. The operation steps summarized in this article are part of secondary development and provide tutorials for beginners.

Acknowledgements

References

[1] **Vijayan S., Sousa Melo S. L., Anamali Allareddy S., Teixeira F. B., Allareddy V.** Segmenting root canal systems using an open source slicer software. Oral Surgery, Oral Medicine, Oral Pathology and Oral Radiology, Vol. 127, Issue 1, 2019, p. 48-49.

[2] **Bruns Nico** 3D slicer: universal 3D visualization software. Der Unfallchirurg, Vol. 122, Issue 8, 2019, p. 662-663.

[3] **Xie Guoqiang, Wang Xin, Hao Wuji, Zuo Yi, Xiao Sanchao, Chen Shangjun** Discussion on the accuracy of diagnosis of hypertensive cerebral hemorrhage hematoma based on 3D Slicer software. International Journal of Neurology and Neurosurgery, Vol. 46, Issue 1, 2019, p. 17-20.

[4] **Shen Ziheng** 3D Slicer10: Architecture1, https://blog.csdn.net/shenziheng1/article/details/76621396.

[5] **Chen Xiaojun, Xu Lu, Wang Huixiang, Wang Fang, Wang Qiugen, Kikinis Ron** Development of a surgical navigation system based on 3D Slicer for intraoperative implant placement surgery. Medical Engineering and Physics, Vol. 41, 2017, p. 81-89.

[6] **Choueib Saleh, Pinter Csaba, Lasso Andras, Fillion-Robin Jean-Christophe, Vimort Jean-Batiste, Martin Ken, Fichtinger Gabor** Evaluation of 3D slicer as a medical virtual reality visualization platform. SPIE Medical Imaging: Image-Guided Procedures, Robotic Interventions, and Modeling, Vol. 10951, 2019, p. 38.

[7]   **Zeng Wenye** 3D Slicer Structure Analysis and Internationalization Method of Medical Image Processing Platform. Zhejiang University of Technology, 2013.
[8]   **Shen Ziheng** 3DSlicer5: Developers know ABC, https://blog.csdn.net/shenziheng1/article/details/76530300.
[9]   **Yu Jianwei, Li Xuwei, Wu Zhiming** Design and implementation of compiler principle demonstration module. Laboratory Research and Exploration, Vol. 37, Issue 9, 2018, p. 138-142+180.
[10]   **Damevski Kostadin, Shepherd David C., Schneider Johannes, Pollock Lori** Mining sequences of developer interactions in visual studio for usage smells. IEEE Transactions on Software Engineering, Vol. 43, Issue 4, 2017, p. 359-371.
[11]   **Zhou Xiaoping, Wang Jia, Guo Ming, Gao Zhe** Cross-platform online visualization system for open BIM based on WebGL. Multimedia Tools and Applications, Vol. 78, Issue 20, 2019, p. 28575-28590.
[12]   **Yang Muyi** Medical image segmentation algorithm based on 3D slicer and its application. Electronic World, Vol. 19, 2016, p. 14-15.

Liyuan Ma is currently studying in the Department of Information and Computational Science, North China University of Science and Technology. Her main research interests include image processing and optimization algorithms.

Chaoqi Zhang is currently an undergraduate of North China University of Science and Technology, who has a strong interest in computer programming. She won the second prize of the Blue Bridge Cup in 2019.

Weina Wu is a student of North China University of Science and Technology, majoring in mathematics and applied mathematics, and working in the 3D modeling innovation laboratory. Her current research interests include research on 3D medical imaging modeling and registration algorithms.

Jincai Chang received his B.Sc. degree in 1996 from Ocean University of China, received his M.Sc. degree in 2005 from Yanshan University, received his Ph.D. degree in 2008 from Dalian University of Technology, now he is Professor in North China University of Science and Technology. His main research interests include theories and methods in mathematical modelling and scientific computation, numerical approximation and computational geometry, etc.

Jianzhong Cui received his Ph.D. degree in 1997 from Tianjin Medical University, now he is Vice President and chief doctor of neurosurgery department of Tangshan Gongren Hospital, and Professor in Hebei Medical University.