

Running Matlab® functions in 3D Slicer using MatlabBridge

Andras Lasso¹, Kevin Alexander², Christopher Jechel², Kevin Wang³, John Schreiner⁴, Gabor Fichtinger¹

¹Laboratory for Percutaneous Surgery, School of Computing, Queen's University; ²Department of Physics, Queen's University; ³Radiation Medicine Program, Princess Margaret Cancer Centre, University Health Network; ⁴Cancer Centre of S.E. Ontario

Research supervisor: Gabor Fichtinger

Background: Matlab® is a very powerful and widely used software environment for development of computational algorithms. However, it is challenging to use Matlab® in medical image computing due to its limited capabilities in medical image data importing/exporting, pre/post-processing, and visualization. 3D Slicer (www.slicer.org) is a free, open source software package that has comprehensive medical image analysis and visualization capabilities and it can be further enhanced and customized by extensions developed in Python or C++ programming languages.

Purpose: Our goal is to implement a software bridge that allows running Matlab® functions directly from 3D Slicer, to: 1) provide rich medical image visualization and pre/post-processing capabilities during Matlab® algorithm development and optimization, and 2) make the developed Matlab® algorithms readily usable for clinicians in the 3D Slicer application. The implementation has to work on all operating systems where 3D Slicer is supported (Windows, Linux, MacOS) and should not require any manual configuration steps or installation of any software packages other than Matlab® and 3D Slicer.

Methods: A new 3D Slicer extension, MatlabBridge was developed, which can start Matlab® and send input data, run the algorithm, and receive processing results using the TCP/IP-based OpenIGTLink protocol (Fig. 1). Each user developed Matlab function has a corresponding MatlabBridge module in Slicer. The graphical user interface of the MatlabBridge module (input/output images, points, scalar value selectors, sliders, editboxes, checkboxes, etc.) are defined in the command-line module descriptor XML format, which is used by numerous medical image computing software applications. The MatlabBridge was used for making Matlab®-based computational algorithms for radiation therapy planning and evaluation accessible in 3D Slicer. Specifically, an implementation of a gamma comparison was evaluated for comparing planned vs. measured dose distributions. The gamma function, introduced by Low *et al.*, is a widely used dose distribution comparison tool which incorporates both dose difference and distance-to-agreement information into a single metric. Dose inputs for the algorithms were imported by the SlicerRT extension (www.slicerrt.org). Standard 3D Slicer volume reslicing and isodose surfaces were used for visualization and analysis of the results.

Results: The MatlabBridge module is available for the latest version of 3D Slicer, it can be installed and configured in a few minutes using the extension manager in 3D Slicer. The extension includes a helper tool (*MatlabModuleGenerator* module) for generating skeleton modules that the developer can customize and extend. The MatlabBridge also allows any other 3D Slicer modules (implemented in Python or C++) to run any Matlab® functions or commands (*MatlabCommander* module). User interface of a 3D gamma Matlab® function and the result of a dose comparison are shown in Fig. 2. No modifications of the original Matlab® functions were needed to run them in 3D Slicer.

Conclusion: Using the MatlabBridge extension, functions implemented in Matlab® can be run from 3D Slicer using a convenient graphical user interface. Practical usability has been demonstrated on computational methods for radiation therapy. Detailed documentation, examples are available at: <http://www.slicer.org/slicerWiki/index.php/Documentation/4.3/Extensions/MatlabBridge>

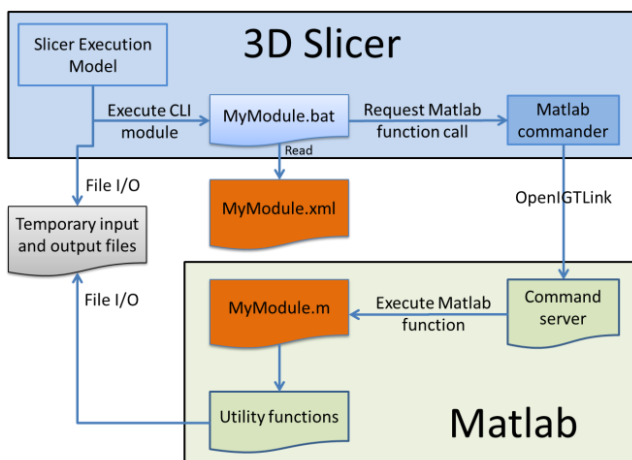


Fig. 1: Architecture overview of the MatlabBridge extension. 3D Slicer and Matlab® communicate through file input/output and OpenIGTLink commands.

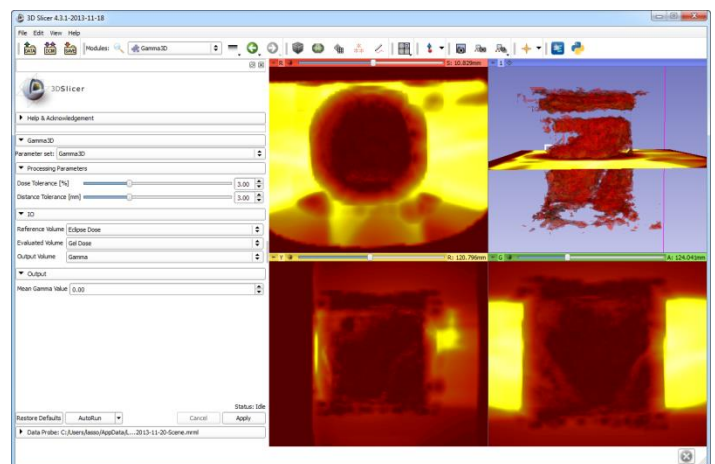


Fig. 2: Comparison result of dose volumes computed by the treatment planning system (Eclipse™) for a prostate IMRT treatment and a gel dosimeter measurement of the dose delivery read by optical CT.

Acknowledgement: This work was in part funded by Cancer Care Ontario through Applied Cancer Research Unit and Research Chair in Cancer Imaging grants and by the CIHR.