# Medical image processing tools for Blender with HPC support

Petr Strakoš*, Milan Jaroš,
Tomáš Karásek, Marta Jarošová,
Daniel Krpelík, Alena Vašatová

IT4Innovations, VŠB-Technical University
of Ostrava, Czech Republic

*petr.strakos@vsb.cz

Tereza Burešová, Juraj Timkovič,
Jan Štembírek, Petr Vávra

The University Hospital in Ostrava,
Czech Republic

## ABSTRACT

We present a medical image processing plug-in in this paper. Our plug-in uses Blender's environment and adds tools for the medical image processing and 3D model reconstructions and measurements. There are several software solutions to provide these tasks, two of which are used for comparison with our tool in this paper. However, there is not a single application that can perform medical image processing and subsequent 3D model reconstruction and post-processing with possible utilization of High Performance Computing (HPC) resources. In our plug-in we aim to use HPC resources to speed up the most computationally extensive tasks. The plug-in is written in Python and C++. The basic functionality and additional tools are implemented in Phyton. Blender has built-in Python interface, thus the development of the tools is quite fast. The C++ language is used for all time critical functions as it has unrestricted potential in parallel programming that can utilize available HPC resources. In the paper we present two examples, where we demonstrate possibilities of our Blender plug-in. We provide 3D reconstruction of human airway and human orbital floor and compare the results with those provided by 3D Slicer and Osirix.

## Keywords
Blender, Python, tools, medical imaging, HPC, parallel computing.

## 1 INTRODUCTION

The fields using powerful computer technologies are getting larger with their availability. Medicine is one of the areas of human activities where such technologies are widely used. Modern sophisticated medical equipment is usually based on a computer technology. As a typical example, Computed Tomography (CT) or Magnetic Resonance Imaging (MRI) scanners can be mentioned. These devices use computer technology to process and provide digital image information of a patient's inner body structure. Such information about individual patients provides a higher level of diagnostic medicine. To move the possibilities even further, the effort is put to the post-processing of acquired data. Image data can be used to provide virtual models of human organs and other tissues. To perform this, besides the powerful hardware, the capable software is necessary.

There are several software solutions specialized in medical image visualization and further processing. Here, we can mention 3D Slicer [4] or Osirix [9] as good examples of long term developed software projects.

3D Slicer is medical image processing software offering tools for the whole image processing cycle, from loading image data stored in the Digital Imaging and Communications in Medicine standard (DICOM), processing the data, to creating 3D reconstructions of selected parts of a human body. Several measurement and result evaluation tools such as distance, area, and volume measurement tools are available.

The other medical image processing software, Osirix, also offers the whole process from loading image data to final 3D reconstructions of selected parts of a human body. Osirix offers several measurement tools for data analysis and result evaluation as well.

Our aim in this paper is to present our own image processing software and introduce its powerful capabilities. Since a lot of image processing methods and techniques is very time consuming, it would be advantageous to have software which can exploit the latest powerful HPC resources. With this in mind, we have started to develop our own medical image processing software implemented as a plug-in for Blender, the 3D

graphics and animation software [1]. Blender is interesting for its Application Programming Interface (API), which allows advanced Python scripting.

Our software is implemented to take advantage of parallel computing and exploit the latest powerful HPC resources. More demanding tasks are implemented in parallel using OpenMP and MPI in C++ language. Our implementation can exploit the computational power of both non-accelerated nodes as well as Intel Xeon Phi accelerated compute nodes of a cluster.

## 2  MODEL RECONSTRUCTION

To provide 3D model reconstructions from medical image data, the software has to go through several typical steps that can be considered as general steps for any software of such kind. We give the explanation as a description of our plug-in components.

First of all, the data has to be loaded. Concerning medical image data, it is stored in the DICOM format. To load this data into Blender, Grassroots DICOM (GDCM) C++ library is integrated in our plug-in.

After loading the medical image data, its visualization is provided to users in a typical axial, sagittal, and coronal view. It might be beneficial to restrict such image data to only some regions of user interest and thus reduce the computational time of the further applied image processing techniques. In our plug-in, we can restrict the computations only to the volume area of our interest by using the box or sphere cutting tool.

Another step within the 3D model reconstruction process is image filtering. The imaging techniques like CT or MRI produce residual image noise, which can have significant impact on subsequent operations performed with an image. To decrease the adverse effect of noise, we have implemented several filtering techniques such as the simple Gaussian smoothing [10], a more sophisticated anisotropic diffusion filter [8], and a very effective and state-of-the-art Sparse 3D Transform-Domain Collaborative filter (BM3D) [3].

After filtering the images, image segmentation is performed. It serves as image simplification in order to localize the objects and their boundaries. Two methods are currently implemented in our plug-in such as a simple image thresholding [10] and k-means clustering [10]. The parallel implementation of the k-means method by using Intel Xeon Phi was introduced in the paper [11]. Both methods can be used separately or combined together.

If the segmentation of the selected body tissue is finished, a reconstruction technique that provides iso-surfaces as a result can be applied. In our plug-in we use Screened Poisson Surface Reconstruction [6] and Metaballs [2]. Both methods use information about identified boundaries as an input parameter.

To create a polygonal mesh from iso-surfaces, the Marching Cubes method is used [7]. This method is time demanding, and so the efficient parallel implementation was necessary. For visualization, Blender Cycles engine is used. To accelerate the computations, the architecture of the Graphic Processing Unit (GPU) (the implementation is already included in Cycles) and Intel's Many Integrated Core (MIC) architecture can be employed. Our implementation for MIC architecture contains all features of the engine with improved performance [5].

The appearance of the plug-in within Blender environment is shown in Figure 1.

## 3  AUXILIARY TOOLS

When the 3D model is reconstructed, some additional software functionality, which can help in diagnostic and planning of the patient treatment, could be required. In the current version of our plug-in, some painting, erasing, cutting, and measuring tools are implemented. A lot of them are based on simple calculations.

We have implemented a tool to correct the results of the semi-automatic segmentation. It is called 2D PAINTER/ERASER and it can also be used to perform manual segmentation. The tool uses a typical brush and erase buttons, which open the segmentation layer for editing.

A similar tool that allows the user to mark the surface area of a selected 3D object but that serves another purpose than image segmentation is called 3D PAINTER/ERASER. This tool works as an intelligent 3D brush, which can paint on 3D surfaces. The surface area marked by the 3D PAINTER/ERASER is then processed by another tool to obtain the total area size of the marked surface. The algorithm behind the tool sums the size of all coloured triangles. The mesh of triangles creates surface of the reconstructed model.

Since surgeons are interested in using this software for planning surgeries, where, e.g., precise measurement of organ volume or resection of a specific part of a tissue is necessary for successful treatment, we have developed two specific tools for such manipulations. The first one is the CUT VOLUME tool for creating arbitrary cuts of a segmented object volume. It provides an infinite plane, which is used as the cutting tool to restrict spatially segmented data.

The second tool calculates the volume of an object and it is named the VOLUME MEASUREMENT tool. The algorithm sums the volumes of each voxel in the object.

Inasmuch as surgeons and radiologists usually work with 2D images, the 3D models resulting from the reconstructions can sometimes be confusing for them. For this reason and also for checking the correctness of the provided reconstructions, we have developed the
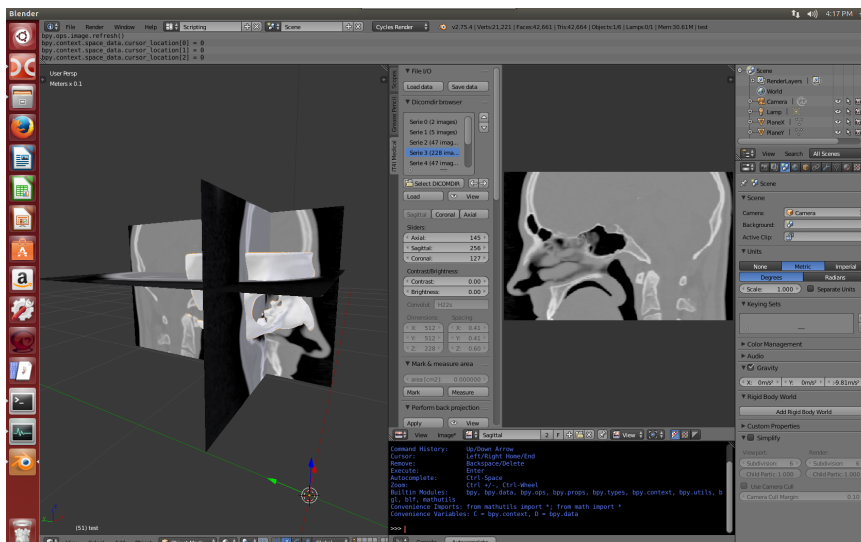
Figure 1: Our plug-in for medical image processing within Blender environment

BACKWARD PROJECTION tool. This tool can show the shape of the reconstructed 3D model as a projection to the original CT data. This allows users to see all important features of the data and at the same time compare accuracy of the provided reconstruction.

## 4  APPLICATION OF THE TOOLS

To show the capabilities of our plug-in, we present here two medical test cases, in which the different functions of our plug-in are exploited. Specifically, we aimed to test precision of the whole 3D reconstruction concept. As a result, we provided 3D models, the volume of which we then measured along with the size of the selected surface area. Within the tests we compared the results performed by our plug-in with the results obtained by the other two already mentioned image processing software solutions, Osirix and 3D Slicer.

For the first test, we selected reconstruction of the human airway, the volume of which was then evaluated. The process of model reconstruction with volume measurement was performed in each software individually. The final results of the volume measurements were then compared.

In the second test, the model reconstruction of the human orbit was done. In each software we carried out the reconstruction from the input data up to the 3D model. As neither Osirix nor 3D Slicer have a tool for measuring surface area marked on the reconstructed model, we exported the models in STL format to Blender, using our tool to mark the surface of the orbital floor and performing the measurement of the selected area. Again, the final results were then compared.

### 4.1  Reconstruction of the airway

Precision of 3D model reconstruction in terms of total volume is very important for medical doctors who will use those models for diagnostic purposes or during surgery planning. For example, while diagnosing sleep apnea, the volume of upper respiratory tract is needed. To mention another example, while planning a liver resection, it is important to calculate the exact size of the liver with lesion, which will be removed during surgery. In both cases, precise information based on 3D model reconstruction is a critical issue.

To compare volumes of 3D models obtained from 3D Slicer, Osirix, and our Blender plug-in, 3D reconstruction of the airway was performed. This was done on DICOM image series of one patient. The density of the data measured by mutual distance between the slices was 0.75mm.

The numerical results showing the airway volume for each software are written in the Table 1. From the results we can see that our model creation concept offers similar model reconstruction as specified by the amount of the final volume, as those measured by other two software.

| Patient | N003 |
|---|---|
| Osirix [ml] | 18.07 |
| 3D Slicer [ml] | 18.33 |
| Our plug-in [ml] | 18.22 |

Table 1: Volume of the reconstructed human airway

### 4.2  Reconstruction of the orbital floor

Another feature requested from medical doctors is the measurement of an area or surface size. During the treatment of orbital fractures, the information about the size of orbital floor damage is critical for the decision about the type of treatment. Our image processing plug-in can provide the information about the size of the injury in terms of surface area measurement. Based on

this finding, the doctors can make the decision whether to provide a conservative or surgical treatment of the patient.

We performed 3D reconstruction of the orbit. This was done on the DICOM image series of three different patients. The density of the data measured by mutual distance between the slices was 0.6mm in all cases. The comparison of the 3D reconstructions provided by 3D Slicer, Osirix, and our Blender plug-in was made. Individual 3D reconstructions from Osirix and 3D Slicer were, due to the lack of a specific measuring tool, exported as STL models and loaded into Blender. Then the orbital floor area was marked and measured by a trained person. For marking and measurement of such surface area, we used our developed tool as described in Section 3.

The numerical results showing the area of the orbital floor surface for each patient and each software are written in the Table 2. The results show that model precision is similar. Moreover, using our tools, we can provide area measurements on reconstructed surfaces, which is a feature available only in our plug-in.

| Patient | N001 | N002 | N010 |
|---|---|---|---|
| Osirix [cm$^2$] | 16.38 | 10.63 | 10.15 |
| 3D Slicer [cm$^2$] | 16.14 | 10.37 | 10.24 |
| Our plug-in [cm$^2$] | 16.28 | 10.49 | 10.14 |

Table 2: Area of the orbital floor for individual reconstruction and patient data

## 5 CONCLUSION

In the presented paper, we have introduced our Blender plug-in equipped with a number of tools for medical image processing. By using it, users can create 3D reconstructions of selected human body tissues from loaded DICOM data. Several tools in the plug-in can utilize the computational support of HPC and parallelization and thus significant processing speed-up can be achieved.

The created 3D reconstructions are of the similar precision as those created in the other two software (3D Slicer, Osirix) we used as a reference. As a unique thing in our plug-in, we provide users with the surface selection and measurement tool for area measurement on the reconstructed surfaces. This can be highly beneficial when, e.g., evaluating the surface area of the orbital floor for studies focusing on eye injuries.

## 6 ACKNOWLEDGMENTS

## 7 REFERENCES

[1] blender.org - Home of the Blender project - Free and Open 3D Creation Software. Available from: <https://www.blender.org>. [18 February 2016].

[2] J. Bloomenthal, C. Bajaj, J. Blinn, M. Gascuel, A. Rockwood, B. Wyvill, G. Wyvill, Introduction to Implicit Surfaces, Morgan Kaufman Publishers Inc, 1997.

[3] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image Denoising by Sparse 3D Transform-Domain Collaborative Filtering, In: IEEE Trans. Image Process., vol. 16, no. 8, pp. 2080–2095, (2007).

[4] A. Fedorov, R. Beichel, J. Kalpathy-Cramer, J. Finet, J-C. Fillion-Robin, S. Pujol, C. Bauer, D. Jennings, F. Fennessy, M. Sonka, J. Buatti, S.R. Aylward, J.V. Miller, S. Pieper, R. Kikinis, 3D Slicer as an Image Computing Platform for the Quantitative Imaging Network, Magnetic Resonance Imaging 2012; July PMID: 22770690.

[5] M. Jaros, L. Riha, T. Karasek, P. Strakos, A. Vasatova, M. Jarosova, T. Kozubek, Acceleration of Blender Cycles Path-Tracing Engine using Intel Many Integrated Core Architecture, Proceedings of the 14th International Conference on Computer Information Systems and Industrial Management Applications, Warsaw, Poland, 2015.

[6] M. Kazhdan, H. Hoppe, Screened Poisson surface reconstruction, ACM Trans. Graphics, 32(3), 2013. (Presented at SIGGRAPH 2013).

[7] W. E. Lorensen, H. E. Cline, Marching Cubes: A high resolution 3D surface construction algorithm, Computer Graphics, Vol. 21, Nr. 4, July 1987.

[8] P. Perona, J. Malik, Scale-space and Edge Detection Using Anisotropic Diffusion, In: IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 629–639. (1990).

[9] A. Rosset, L. Spadola, O. Ratib, OsiriX: an open-source software for navigating in multidimensional DICOM images, J Digit Imaging, 2004 Sep;17(3):205-16, Epub 2004 Jun 29; PMID: 15534753.

[10] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Thomson, (2006).

[11] P. Strakos, M. Jaros, T. Karasek, L. Riha, M. Jarosova, T. Kozubek, P. Vavra, T. Jonszta, Parallelization of the Image Segmentation Algorithm for Intel Xeon Phi with Application in Medical Imaging, in P. Ivnyi, B.H.V. Topping, (Editors), Proceedings of the Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Civil-Comp Press, Stirlingshire, UK, Paper 7, 2015.