Vessel enhancement and segmentation visualizer

Renzo Phellan[1]

[1]Department of Radiology and Hotchkiss Brains Institute, University of Calgary,
Calgary, AB, Canada
phellan.renzo@ucalgary.ca

Abstract. Many applications have been developed to process and visualize medical images. However, they are designed to handle many different tools. Thus, the user may need more time to learn how to use them properly and get adapted to a general user interface. Vessel enhancement algorithms are used to improve the accuracy of blood vessel segmentation in medical images of different modalities. Currently, there is no particular application addressed to analyze and compare three dimensional medical images of the brain and their corresponding segmentations after vessel enhancement. In this context, I decided to develop a simple and specific application to overcome that issue. In this paper, I describe the main elements of the new application and the tools I used to develop it.

Keywords: Vessel enhancement, vessel segmentation, brain vascular system visualization.

1 Introduction

Segmentation of the cerebrovascular system in 3D medical images can support clinicians in various medical situations, e.g. diagnosing cerebrovascular malformations [1], determining brain tumor malignancy [2], and planning surgical and interventional radiology procedures [3].

Manual vessel segmentation is known to be a time-consuming, error-prone task and subject to inter-observer variability [4, 5]. As a consequence, automatic segmentation of the brain vascular system is a very active research field.

Within this context, vessel enhancement algorithms are often applied to improve the results of vessel segmentations from various image modalities and organs, e.g. MR images of the brain [4], CT images of the liver [6], and CT images of the lungs [7]. Vessel enhancement can be conceptualized as a preprocessing step, where voxels belonging to vessels are enhanced in comparison to other voxels.

Recently, many applications have been developed to visualize 3d medical images, such as ITK-SNAP [8], 3D Slicer [9] and OsiriX [10]. However, they are designed to be general applications that include many different tools for medical image processing and visualization. The abundance of processing and visualization elements may confuse the user and lengthen the process of learning how to use the application. Besides, a user interface designed to support different tools may not effectively address all of them.

As a consequence, I decided to develop an application particularly addressed to analyze and compare three dimensional medical images and their corresponding segmentations after vessel enhancement. It is a simple application with enough specific elements for vessel enhancement visualization and it is described in the following lines.

2 Principal components

The main screen of the application is shown in Figure 1. It can be divided in four main sections. Section A contains tools to visualize the original medical image, without enhancement. Section B contains tools to visualize the enhanced medical image, after applying a vesselness enhancement algorithm. Section C contains a set of buttons to load the images which are being analyzed and their corresponding segmentations. Section D is designed to present quantitative metrics to compare the original and enhanced image segmentations. The components of each section are detailed in the following subsections.

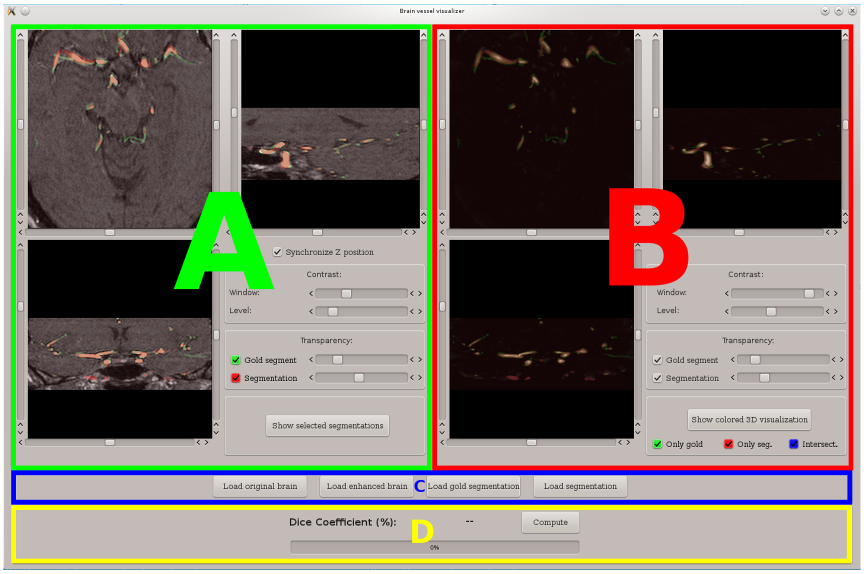
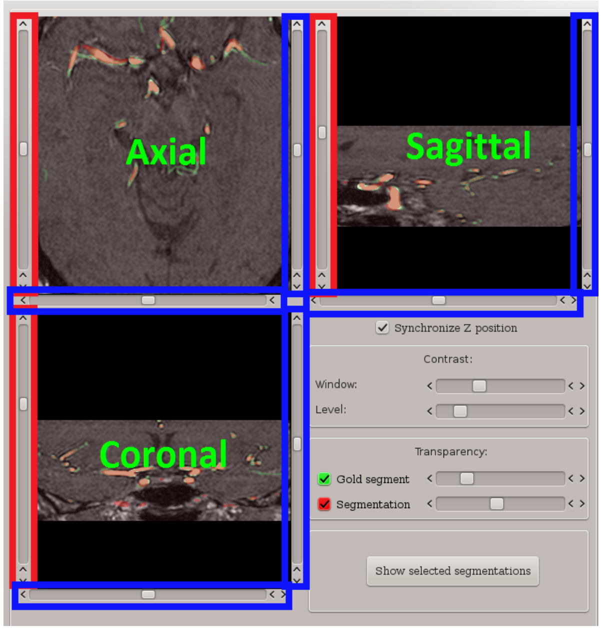


Fig. 1. Main screen of the vessel enhancement and segmentation visualizer.

2.1 Section A: Original image

Section A is intended to display the original medical image, without enhancement. As it can be seen in Figure 2, the axial, sagittal and coronal slices of the three dimensional image are displayed in Section A. It is possible to navigate through the slices by using the vertical bars indicated in red in Figure 2. The slice can be moved in the left-right or up-down directions using the respective horizontal and vertical bars indicated in blue.



☑ Synchronize Z position

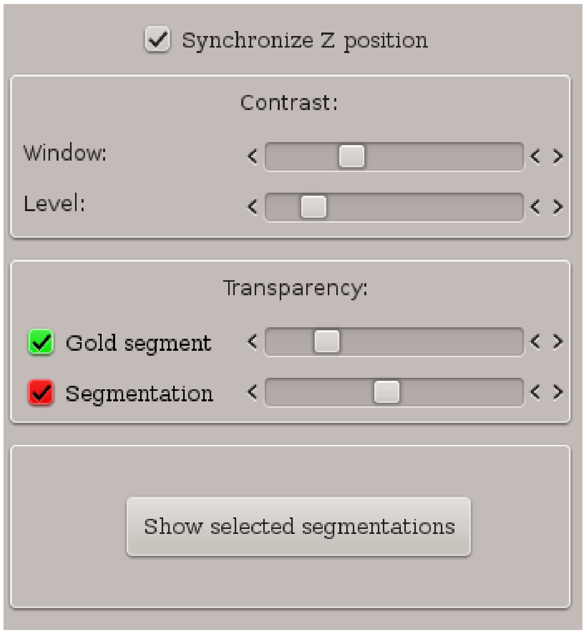
☑ Gold segment

☑ Segmentation

Fig. 2. Section A: Original image.

There is also a control panel below the Sagittal view. It is zoomed in Figure 3. The ***Synchronize Z position*** option, allows the user to synchronize the vertical bars indicated in red in both sections, A and B. That way, the user can navigate through the slices of corresponding views in both sections, simultaneously. The ***Contrast*** frame contains two horizontal bars that allow the user to adjust the Window and Level of the image displayed in Section A. The ***Transparency*** frame contains two selection boxes that control the opacity of the segmentations overlaid on the original image in Section A. It also contains two check boxes that let the user hide or show the corresponding segmentation mask. The color of the box corresponds matches the color of the segmentation on the image. Finally, the ***Show selected segmentations*** button, creates a three dimensional representation of the segmentations that are selected in the Transparency frame.

The three dimensional representation of the segmentations that are selected in the Transparency frame is interactive. In Figure 4, it can be seen that the volumes are inside a box. Each ball point of the box can be used to move the plane that contains that point and cut the volumes using the plane. This characteristic allows the user to focus on a specific region of interest. The interactive box can



☑ Synchronize Z position

☑ Gold segment

☑ Segmentation

Fig. 3. Section A: Control panel

be hidden by pressing the key ***i***. Without the box, the volume can be rotated or zoomed using the mouse buttons and wheel, respectively.

2.2 Section B: Enhanced image

Section B is intended to display the enhanced medical image, which results after applying a vessel enhancement algorithm. As it can be seen in Figure 5, the axial, sagittal and coronal slices of the three dimensional image are displayed in Section B. It is possible to navigate through the slices by using the vertical bars indicated in red in Figure 5. The slice can be moved in the left-right or up-down directions using the respective horizontal and vertical bars indicated in blue.

There is also a control panel below the Sagittal view. It is zoomed in Figure 6. The ***Contrast*** frame contains two horizontal bars that allow the user to adjust the Window and Level of the image displayed in Section B. The ***Transparency*** frame contains two selection boxes that control the opacity of the segmentations overlaid on the original image in Section B. It also contains two check boxes that let the user hide or show the corresponding segmentation mask. The color of the box corresponds matches the color of the segmentation on the image. Finally, the ***Show colored 3D visualization*** button, creates a three dimensional representation of the segmentations.

The three dimensional representation of the segmentations in section B is different from the one presented in Section A. In Figure 7, you can see three

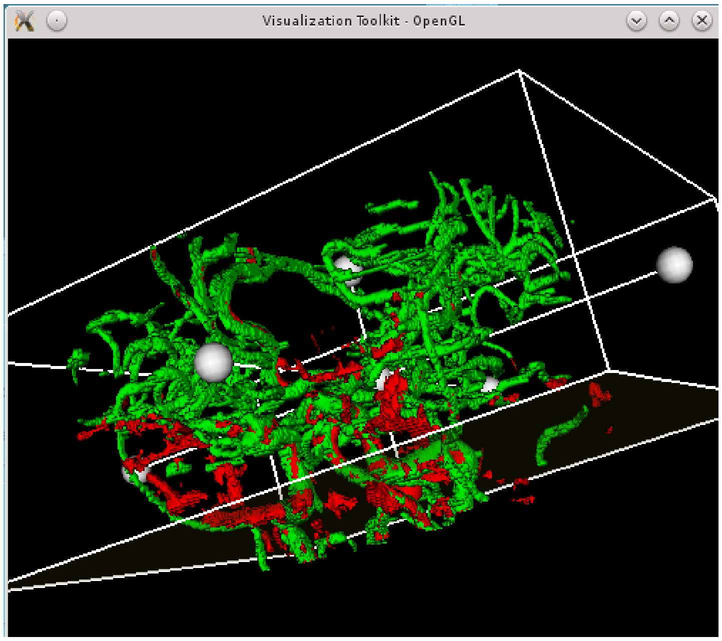


Fig. 4. Section A: Three dimensional view of segmentations

different colors: red, green and blue. They match the three selection boxes which are below the ***Show colored 3D visualization*** button. In that context, the red points belong exclusively to the segmentation generated after enhancement, the green points belong exclusively to the segmentation generated before enhancement, and the blue points are common to both segmentations. All three sets of points can be hidden by unselecting the corresponding selection box.

2.3 Section C: Data loading

Section C has four buttons, as shown in Figure 8. They allow the user to load the data which is going to be visualized: ***Load original brain*** to load the original brain without enhancement, ***Load enhanced brain*** to load the brain enhanced after applying a vessel enhancement algorithm, ***Load gold segmentation*** to load the segmentation corresponding to the original brain, and ***Load segmentation*** to load the segmentation corresponding to the enhanced brain. Data should be loaded following the order of the buttons to guarantee a proper functioning of the application. In general, the original and enhanced brains should be loaded before their corresponding segmentations.

2.4 Section D: Quantitative metrics

Section D is the last section and is designed to present quantitative metrics to compare the segmentations corresponding to the original and enhanced medical

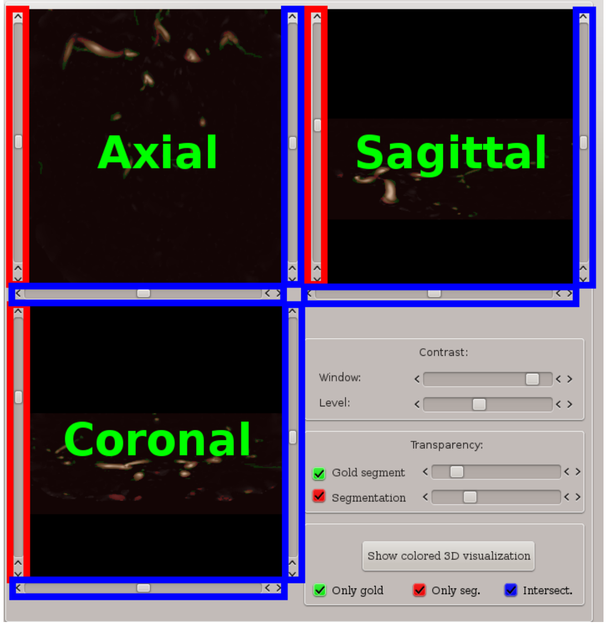


Fig. 5. Section B: Enhanced image.

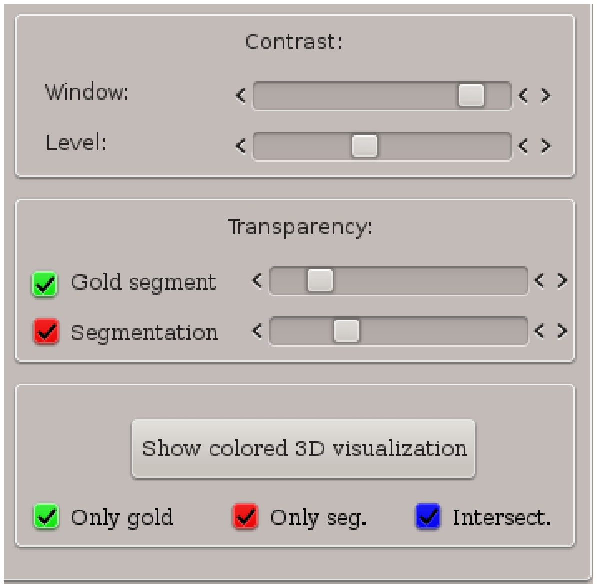


Fig. 6. Section B: Control panel

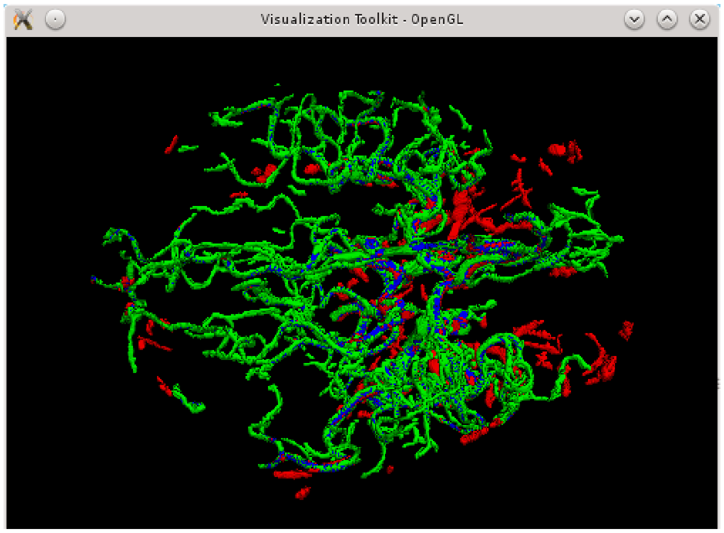


Fig. 7. Section B: Three dimensional view of segmentations

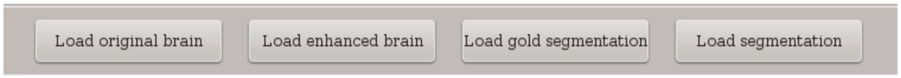
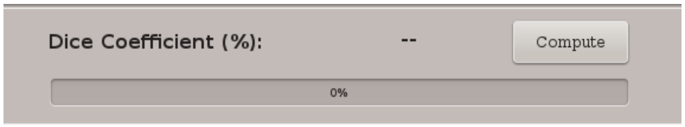


Fig. 8. Section C: Data loading

images and it can be seen in Figure 9. The selected metric is the Dice Coefficient [11]. Essentially, it measures the amount of overlapping between two different segmentations. This criterion has been proven to be useful to evaluate segmentations [12] when compared to others like True Positive or True Negative Volume Fraction [13].

In order to calculate the Dice Coefficient between segmentations, the user should click the ***Compute*** button. The loading bar in the bottom displays the progress of the operation. The computed value for the Dice coefficient is displayed as a percentage next to the ***Dice Coefficient(%)*** label.


Dice Coefficient (%): --

0%

Fig. 9. Section D: Quantitative metrics

3 Development tools

The whole application was developed using three main tools: the Python programming language [14], the Visualization Toolkit (VTK) framework [15], and QtDesigner [16]. However, other specific software was also used and it is listed in Table 1.

Software	Version	Function
Nibabel [17]	2.0.2	To read Niftii images.
PyQt [18]	4.11.2	To program interfaces in Qt and handle them with Python.
Python [19]	2.7.9	Programming language.
Pyuic [20]	4.0	To translate the Qt code of QtDesigner into Python code.
Qt [21]	4.8.6	The programming language for user interfaces.
QtDesigner [22]	4.8	A graphical interface designing tool based in Qt.
VTK [23]	5.8.0	Image processing and visualization algorithms.

Table 1. Software used to develop the application presented in this paper.

4 Future work

The present application can be improved by adding more characteristics to the interface. All slice views should allow the user to zoom in and out of the image. Besides, the colors corresponding to every segmentation may be chosen by the user. New metrics can be added to Section D as they are required.

Regarding the image renderings, more detailed images can be generated by using recent algorithms and by leveraging the computing power of graphical processing units. In this application, rendering is done by using the Marching Cubes algorithm [24], followed by a Decimation step [25] to reduce the number of triangles generated by Marching Cubes algorithm and accelerate the response of the application to user interaction.

5 Conclusion

The present paper describes an application to visualize and compare three dimensional images of the brain, before and after processing them with a vessel enhancement algorithm. Manual and automatic segmentations of vessels can be overlaid on the images. Additionally, specific tools are provided to navigate and adjust the images for a clearer visualization. Three dimensional renderings of both segmentations are also available.

References

[1] Forkert, N.D., Fiehler, J., Illies, T., Möller, D.P., Handels, H., Säring, D.: 4D blood flow visualization fusing 3D and 4D MRA image sequences. Journal of Magnetic Resonance Imaging **36**(2) (2012) 443–453

[2] Bullitt, E., Zeng, D., Gerig, G., Aylward, S., Joshi, S., Smith, J.K., Lin, W., Ewend, M.G.: Vessel tortuosity and brain tumor malignancy: a blinded study. Academic radiology **12**(10) (2005) 1232–1240

[3] Aylward, S.R., Bullitt, E.: Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. Medical Imaging, IEEE Transactions on **21**(2) (2002) 61–75

[4] Forkert, N.D., Schmidt-Richberg, A., Fiehler, J., Illies, T., Möller, D., Säring, D., Handels, H., Ehrhardt, J.: 3D cerebrovascular segmentation combining fuzzy vessel enhancement and level-sets with anisotropic energy weights. Magnetic resonance imaging **31**(2) (2013) 262–271

[5] Phellan, R., Falcão, A.X., Udupa, J.K.: Medical image segmentation via atlases and fuzzy object models: Improving efficacy through optimum object search and fewer models. Medical physics **43**(1) (2016) 401–410

[6] Luu, H.M., Klink, C., Moelker, A., Niessen, W., van Walsum, T.: Quantitative evaluation of noise reduction and vesselness filters for liver vessel segmentation on abdominal CTA images. Physics in medicine and biology **60**(10) (2015) 3905

[7] Rudyanto, R.D., Kerkstra, S., Van Rikxoort, E.M., Fetita, C., Brillet, P.Y., Lefevre, C., Xue, W., Zhu, X., Liang, J., Öksüz, İ., et al.: Comparing algorithms for automated vessel segmentation in computed tomography scans of the lung: the VESSEL12 study. Medical image analysis **18**(7) (2014) 1217–1232

[8] Yushkevich, P.A., Piven, J., Cody Hazlett, H., Gimpel Smith, R., Ho, S., Gee, J.C., Gerig, G.: User-guided 3D active contour segmentation of anatomical structures: Significantly improved efficiency and reliability. Neuroimage **31**(3) (2006) 1116–1128

[9] Fedorov, A., Beichel, R., Kalpathy-Cramer, J., Finet, J., Fillion-Robin, J.C., Pujol, S., Bauer, C., Jennings, D., Fennessy, F., Sonka, M., et al.: 3d slicer as an image computing platform for the quantitative imaging network. Magnetic resonance imaging **30**(9) (2012) 1323–1341

[10] Rosset, A., Spadola, L., Ratib, O.: Osirix: an open-source software for navigating in multidimensional dicom images. Journal of digital imaging **17**(3) (2004) 205–216

[11] Dice, L.R.: Measures of the amount of ecologic association between species. Ecology **26**(3) (1945) 297–302

[12] Yeghiazaryan, V., Voiculescu, I.: An overview of current evaluation methods used in medical image segmentation. (2015)

[13] Udupa, J.K., Leblanc, V.R., Zhuge, Y., Imielinska, C., Schmidt, H., Currie, L.M., Hirsch, B.E., Woodburn, J.: A framework for evaluating image segmentation algorithms. Computerized Medical Imaging and Graphics **30**(2) (2006) 75–87

[14] Sanner, M.F., et al.: Python: a programming language for software integration and development. J Mol Graph Model **17**(1) (1999) 57–61

[15] Schroeder, W., Martin, K., Lorensen, B.: The visualization toolkit: an object oriented approach to 3D graphics. New York: Kitware. Inc. Publisher (2006)

[16] Summerfield, M.: Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming. Pearson Education (2007)

[17] : NiBabel. http://nipy.org/nibabel/ Accessed: 2016-03-29.

[18] : PyQt. https://wiki.python.org/moin/PyQt Accessed: 2016-03-29.

[19] : Python. https://www.python.org/ Accessed: 2016-03-29.

[20] : Jonathan Gardner's PyQt Tutorial. https://wiki.python.org/moin/JonathanGardnerPyQtTutorial/ Accessed: 2016-03-29.

[21] : Qt. http://www.qt.io/ Accessed: 2016-03-29.

[22] : Qt Designer Manual. http://doc.qt.io/qt-4.8/designer-manual.html Accessed: 2016-03-29.

[23] : VTK. http://www.vtk.org/ Accessed: 2016-03-29.

[24] Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: ACM siggraph computer graphics. Volume 21., ACM (1987) 163–169

[25] Schroeder, W.J., Zarge, J.A., Lorensen, W.E.: Decimation of triangle meshes. In: ACM Siggraph Computer Graphics. Volume 26., ACM (1992) 65–70