

OCTSEG software manual

OCTSEG = Optical Coherence Tomography Segmentation and Evaluation GUI

Abstract

The purpose of this document is to describe the algorithm and functionality of the OCTSEG software. This software segments the retinal layers, especially the retinal nerve fiber layer (ONFL) of OCT-volume scans. This document provides information in addition to the original OCTSEG user manual.

Introduction

OCTSEG (Optical Coherence Tomography Segmentation and Evaluation GUI) is a graphical user interface (GUI) written in MATLAB for research purposes. With this software, the retinal layers and the blood vessels of retinal OCT scans can be segmented. Tools for the manual correction of the automated segmentations are provided. The segmentations and resulting thickness measurements on the retinal layers may be exported as a CSV file, which is readable by standard software (e.g. Excel).

OCTSEG software has been originally developed by Markus Mayer, Pattern Recognition Lab, University of Erlangen Nuremberg, Germany (Mayer) and released in 2012. The algorithms used in this software are partly described in (Mayer, Hornegger, Mardin, & Tornow, 2010) and partly in this document. The software has been modified to run on few example data from AMC for the EYR4 “A flightpath for OCT imaging project” (Almasian, 2013) by Elena Rangelova from the Netherlands eScience center (NLeSc) at the end of 2014/beginning of 2015.

This version is available at the NLeSc repository, (Rangelova, 2015), and is the subject to this document. The original `octsegManual.txt` and other related documentation, being referred to in this text, are available at the folder of this document (`OCTSegmentation/help`) unless specified otherwise.

Main Features

Supported data formats

- Circular OCT scans as well as Optic Nerve Head (ONH) centered volumes. The supported data format as Heidelberg Engineering Spectralis OCT RAW data (.vol extension)
- OCT data stored as image files (.tif, .pgm, .jpg) is also supported. Multiple images can be read in as volumes using user generated .list files (a text file format).

Functionality

- Automated segmentation of 6 prominent retinal layers (including the inner limiting membrane, outer nerve fiber layer boundary, and retinal pigment epithelium)
- Automated segmentation of the blood vessel positions on circular scans
- Batch processing of circular scans
- Manual correction of possible segmentation errors
- Visualization of the data and the segmentation results, including enface views and thickness maps
- Export of the segmentation results to CSV text files

Phantom segmentation

Currently the software does not support directly from the GUI segmentation of phantom data (either in CSV or BIN files). Initial version of a separate software using the available algorithms, which could eventually be added to OCTSEG can be found at (OCTSegmentation/PhantomSegmentation).

Algorithms

Part of the algorithms implemented by OCTSEG are described in the original author's publication (Mayer, Hornegger, Mardin, & Tornow, 2010). The main source for understanding the algorithms (OCTSegmentation/algo) is the code itself (Ranguelova, 2015). Here only the main ideas are outlined.

Preprocessing of the volume

The speckle noise of the volume data is reduced by weighted averaging in the 3D space. The subsequent segmentation steps are performed on the individual B-Scans.

Segmentation steps

Six prominent layers are segmented in the order shown in Figure 1. The search space for the boundaries is decreased by each step. The colors corresponding to the layers are: **blue**: retinal pigment epithelium (RPE); **red**: inner limiting membrane (ILM); **yellow**: outer nuclear layer (ONL); **green**: outer plexiform layer (OPL); **orange**: inner plexiform layer (IPL); **cyan**: outer nerve fiber layer boundary (ONFL).

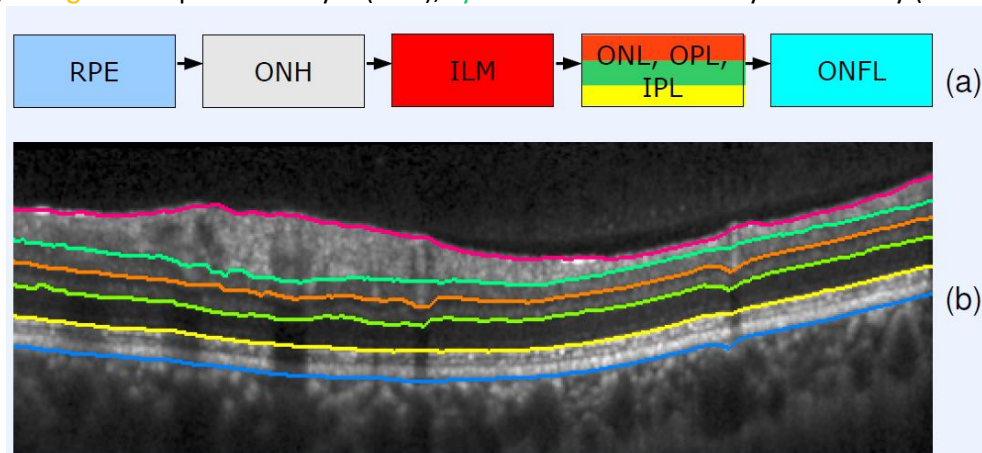


Figure 1 (a) Segmentation steps. (b) B-scan to visualize the order and position of the boundaries.

Segmentation of the layers

The RPE, the ONL, the OPL, the IPL and the ILM are determined by the same algorithm (see Figure 2):

- The B-Scan is denoised by median filtering.
- An edge detection that takes the second derivative into account generates a initial segmentation.
- Using RANSAC (a model fitting method), a polynomial of degree 5 is fitted through the initial segmentation.
- The error measurement used for optimizing the model fit in RANSAC is the L1 norm of the column wise distance between the initial segmentation and the polynomial.
- The points on the initial segmentation that exceed a certain distance from the polynomial are replaced by the points of the polynomial. The distance threshold is dependent on the currently segmented boundary.
- Blood vessel regions are linearly interpolated.

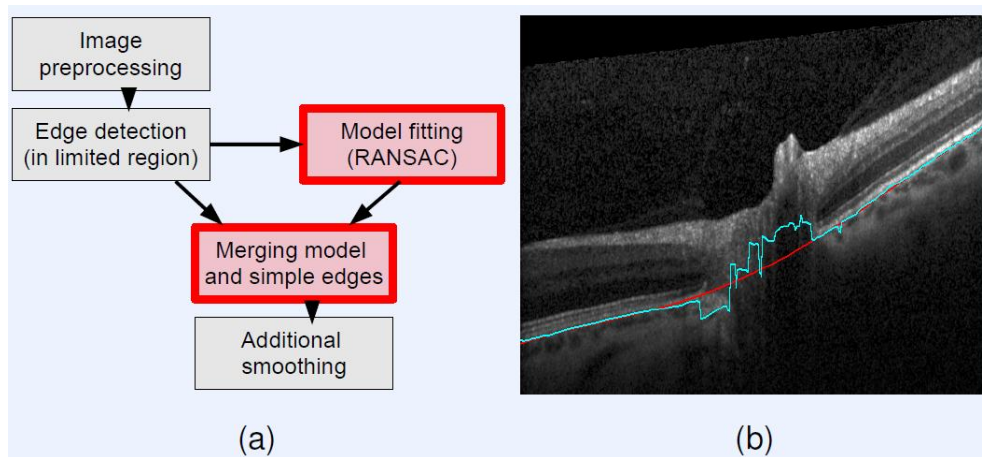


Figure 2 (a) Segmentation algorithm for the layers. (b) Example B-Scan: Edge detection result for the RPE(blue). RANSAC fitted polynomial with L1 norm as error measurement (red).

Segmentation of the ONFL

The B-Scan is denoised by complex diffusion. The ONFL boundary is identified in between the IPL and the ILM using an energy minimization approach (Mayer, Hornegger, Mardin, & Tornow, 2010) that takes the local gradient as well as local smoothness into account.

Postprocessing

A median filtering of the layer boundaries in the 3D space of the volume yields the final results.

Compilation and Distribution

The software has been compiled using MATLAB version 7.6.0.324 (R2008a) with compiler version 4.8 and is freely available as a stand-alone application (Rangelova, 2015). The followed procedure for compiling and packaging for redistribution is according to the **Deploying Matlab application (Windows)** tutorial (KU Leuven) which is also to be found at the current document's folder. Please note, that the version numbers in the tutorial are slightly different than the one mentioned above. To be deployed on a machine without MATLAB installed, one should follow the instruction in the tutorial as of section *Deploying a Matlab application* (KU Leuven), page 8 and also according to the readme.txt file in the distribution folder (OCTSegmentation/Executables/octseg/distrib).

A compiled version of a very initial partial segmentation of 2D CSV phantoms is also available at (OCTSegmentation/Executables/phantomseg/distrib).

Screenshots

These screen shots are given here only as additional illustration of the explanations given in the original software documentation (octsegManual.txt).

The screenshots are from running the program within MATLAB. When using the stand-alone compiled version of the software, the GUI windows have the same appearance, only the command window is the Windows command window (compare Figure 3 and Figure 4).

Figure 3 and Figure 4 show the main window of the GUI. It is used for loading, segmenting, importing and exporting data. For detailed description of the modes, information tables and menu commands, please refer to the original documentation (octsegManual.txt) available also at the current document's folder.

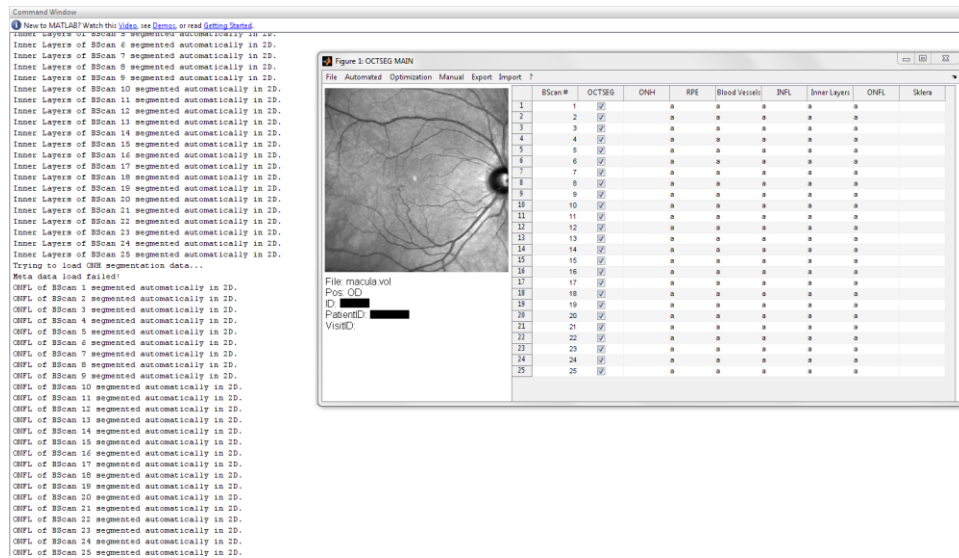


Figure 3 Main window (in MATLAB). Automatic segmentation of Spectralis volume.

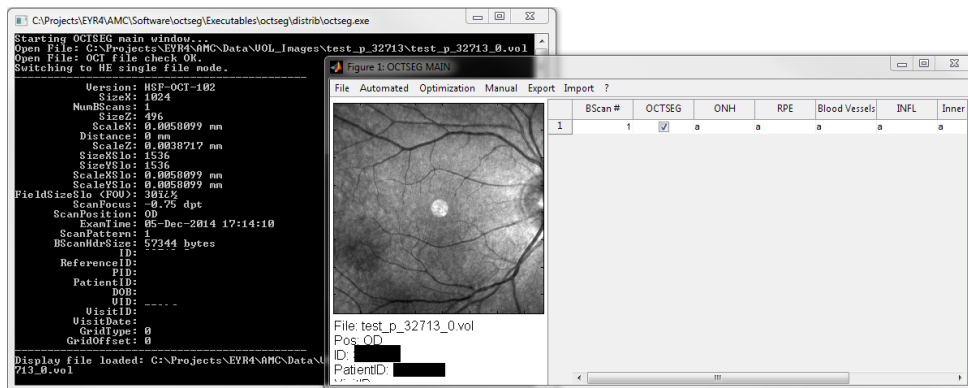


Figure 4 Main window (stand-alone application). Loading of the test VOL images.

Invoking the visualization part (Visu) of the software, a new window opens where the OCT and SLO views with indication of the position of the B-Scan and browsing using a slider is possible. Figure 5 shows the first, fifth and 20th scan of the same volumetric data.

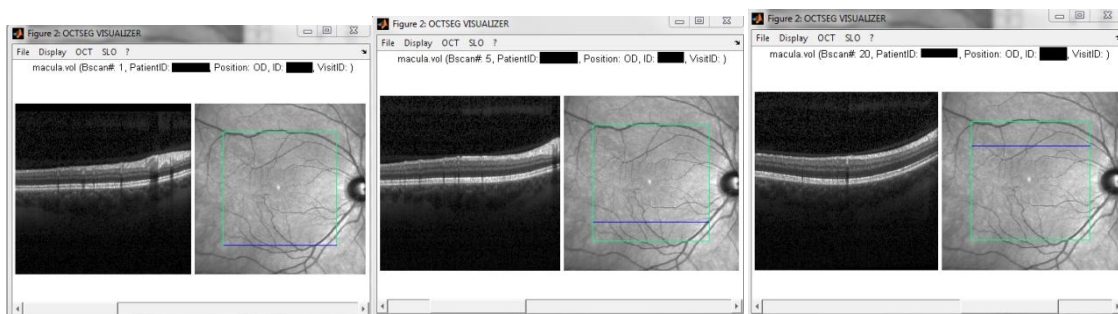


Figure 5 OCT and SLO views of volumetric data

The OCT data are displayed in double sqrt gray scale by default, but it can be changed to sqrt, liner and logarithmic (Figure 6). In Visu the segmentation results are also displayed (Figure 7).

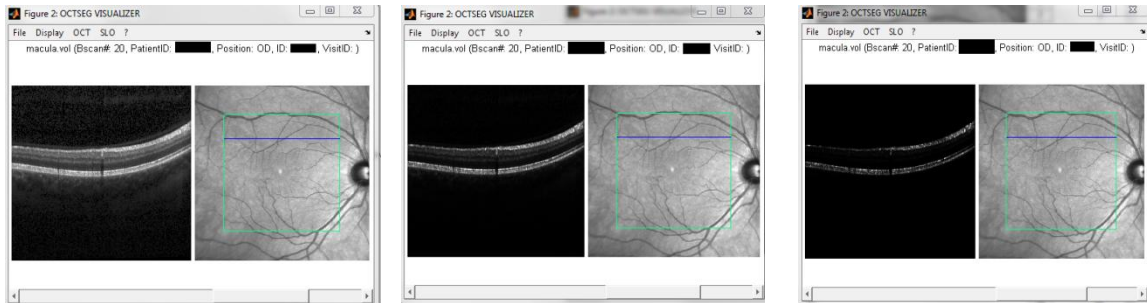


Figure 6 From left to right: double sqrt, sqrt and linear gray scale display.

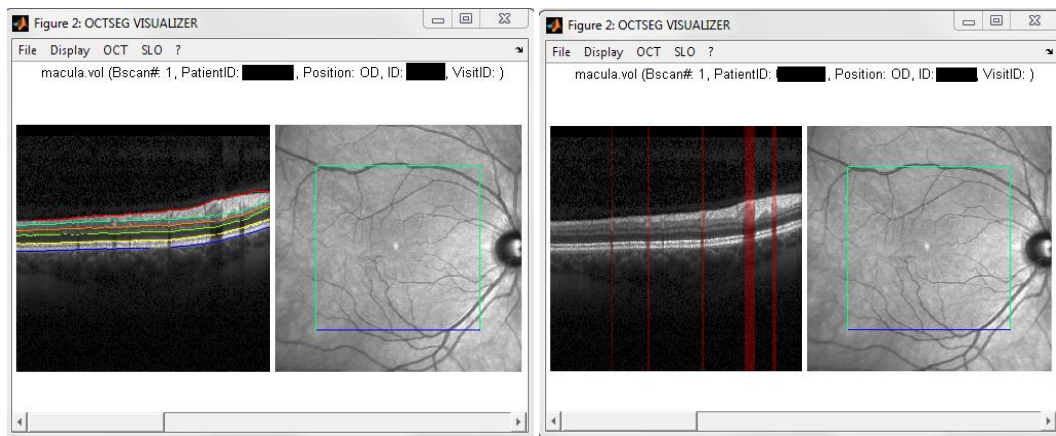


Figure 7 Segmentation of the layers and blood vessels

OCTSEG also offers the possibility of manual correction of the segmentation result, when the automatic result is not satisfactory as shown on Figure 8. The boundary thickness for display can also be chosen.

The software has some extra visualization options, for example displaying thickness maps and enface views overlaid over the SLO view (Figure 9).

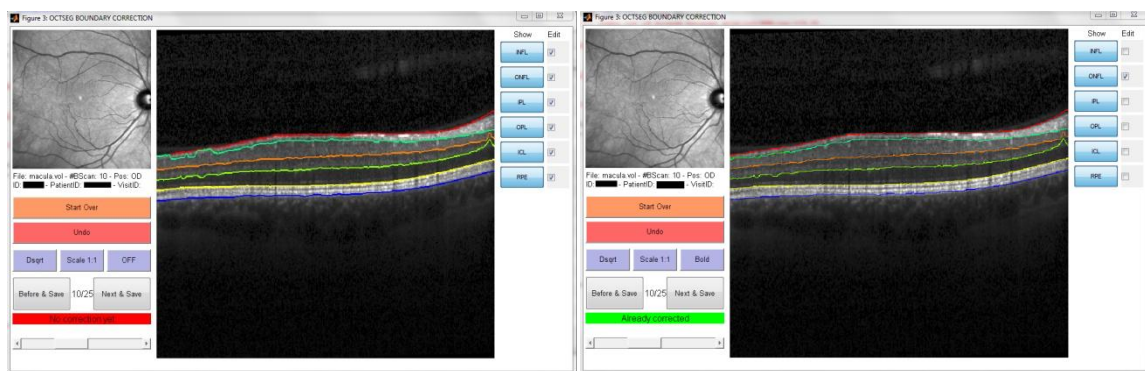


Figure 8 Manual boundary correction: before and after



Figure 9 From left to right: Retina thickness, NFL thickness map and sclera enfance

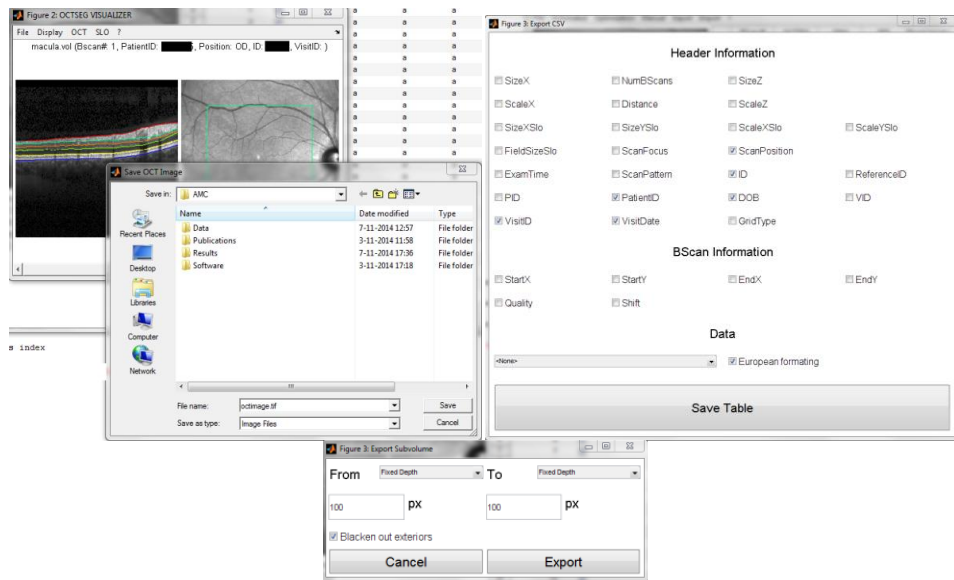


Figure 10 Exporting the segmentation results

Figure 11 illustrates the preliminary work on the phantom segmentation.

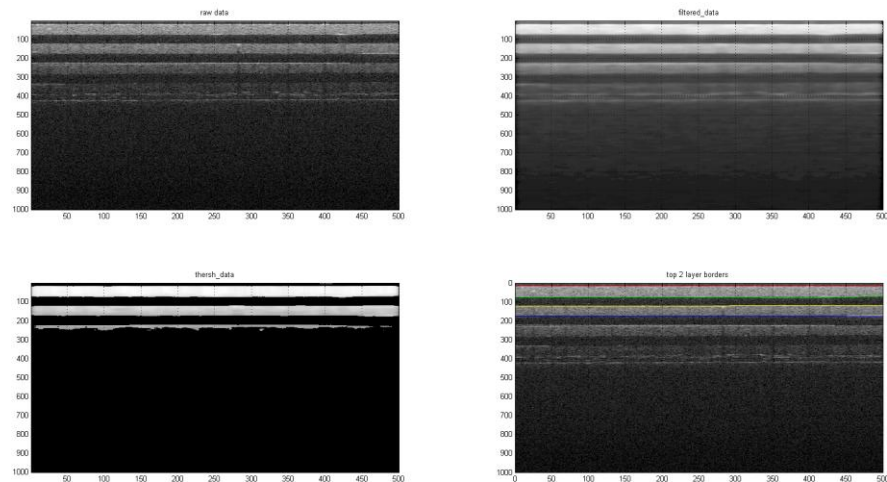


Figure 11 Segmentation of the top layers of a phantom

References

- Almasian, M. (2013, October). *A Lightpath for Optical Coherence Tomography Imaging*. Retrieved from <https://blog.surfnet.nl/?p=2747>
- KU Leuven. (n.d.). *Deploying Matlab application (Windows)*. Retrieved from <https://admin.kuleuven.be/icts/onderzoek/wetsoft/software/matlab/pdf/matlab-deploytool-standalone>
- Mayer, M., Horneegger, J., Mardin, C., & Tornow, R. (2010). Retinal Nerve Fiber Layer Segmentation on FD-OCT Scans of Normal Subjects and Glaucoma Patients. *Biomedical Optics Express*, 1358-1383.
- Meyer, M. (n.d.). *OCTSEG*. Retrieved from OCTSEG: <http://www5.cs.fau.de/en/research/software/octseg/>
- Ranguelova, E. (2015). *OCT Segmentation*. Retrieved from <https://github.com/NLeSC/OCTSegmentation>