Manual - OCT Segmentation

Fabian Rathke

December 18, 2013

1 Description

... is a software package that can be used to segment retina OCT-scans: 2-D B-Scans or 3-D volumes consisting of several B-scans. It is mainly written in Matlab and C and supports the usage of a GPU via the free Matlab interface GPUmat.

2 Installation

Installation in Matlab is accomplished by adding the source code path via

```
addpath('/path/to/source/code');
```

either into the respective Matlab script or for a permanent adding into the startup.m, a script called by Matlab during each startup. Some functions were written in C, to speed up computations. To be able use these files, they have to compiled from within Matlab using the *mex* function. There are three C functions:

```
cd collector/helperFunctions/
mex getRaster.c
cd ../../prediction/predVariational
mex optQCC.c
cd ../../training/helperFunctions
mex cglasso.c
```

Furthermore, several functions have be implemented such that they can be either run on the CPU or on a Nvidia GPU. In order to use this functionality, CUDA has to be installed. Additionally, download the GPUmat package for your platform, extract it and add the path to Matlab.

3 Basic Usage

In order to segment a OCT scan, we need to train appearance models and a shape prior model. To do this we need *ground truth* in form of manually labeled OCT scans. If you have no ground truth available yet, you can use our labeling tool that comes with the segmentation software, see section 5.

Given the trained models we can make predictions for unknown OCT scans. Given ground truth for these scans, we can evaluate different error measures. The subsequent sections will explain each part with example code.

We assume that a data set of n scans is available (either 2-D or 3-D)

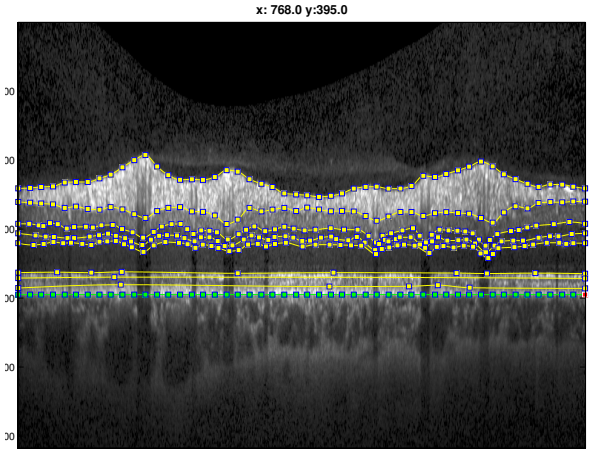

Figure 1: B-Scan with labels provided for 9 boundaries. The scan was flattened along the 9th boundary to ease labeling (press 'r' with the respective boundary selected).

3.1 Training

We first load a set of files and determine folders which contain the data as well as ground truth.

3.2 Prediction

3.3 Evaluation

4 Parameters

5 Labeling Tool

Included in our software package is a labeling tool that allows you to establish ground truth. Fig. 1 shows a B-Scan with 9 boundaries labeled. All options are documented inside the m-file, i.e. call 'help labelTool' for all options available.

5.1 Configuration

```
% the ID of the B-scan
collector.options.labelID = ID;
collector.options.loadRoutineData = 'dataRoutine';
collector.options.folder_data = '/path/to/oct/scans/';
% if restore == 1 and we have already labeled data with the label tool
collector.options.saveDir = '/path/to/save/labels/';
% if restore == 1, we have not labeled anything yet but there are labels provided
    with the dataset
collector.options.loadRoutineLabels = 'loadRoutine';
collector.options.folder_labels = '/path/to/initial/labels';
% we want no preprocessing (we could add here a Gaussian filter for example or any
    user-defined function to ease labeling)
collector.options.preprocessing.scanLevel = {};
% using existing labels?
restore = 1;
% we label the B-Scan with ID stored inside 'filename.mat';
```

```
labelTool('filename.mat',collector,restore);
```

5.2 General Usage

After you call the function you will see the scan and labels already provided, in case you choose to restore them. First you can crop the scan in order to cut away parts of the scan that are of no interest, first the vitreous body, then the choroid. You can skip each step by pressing 'c'.