

CODE EXPLANATION

This chapter explains the code attached to this master thesis, and what is needed to test the different modules of our prototype. There are three modules which will be explained:

- The segmentation module
- The object recognition module
- The CBR module

Each of these modules are explained in the subsequent sections.

SEGMENTATION MODULE

The code¹ for this module is responsible for segmenting the regions of the image, and uses the live-wire approach, as presented in the thesis. To start this module, simply compile it and run the livewire-class. The image to be segmented can then be selected, and the live-wire application starts. To segment the image, simply mark the region you want to segment, and press train image. The cost matrix is then updated, and visualised in the right-hand window of the screen. By pressing reset, the image can be resegmented using the newly created cost matrix. By repeating this process, a better segmentation result is obtained. When the result is satisfactory, the image can be saved, and it is ready for the next module.

OBJECT RECOGNITION MODULE

This module is split in two sections. First, the regions of the segmented image must be extracted and described. This is done by the `extract_regions`-function. This results in a struct containing the information about the regions present in the image. This struct must then be saved to an XML-file, before matched against the organ database. A small subset of the organ database is contained in the `MAD_reg_ref.xml`-file. The actual matching is performed by using the `match_regions`-function, with the stored struct (XML-file) and the organ database as input. The result of this matching procedure is a new XML-file which is composed in a form suitable for the CBR module.

CBR MODULE

The CBR module must load a predefined domain knowledge model before analysing the particular standard view plane. When this model has been loaded, the XML-file describing the findings of the standard view plane can be loaded for analysis. This requires the use of the `jCreek` libraries.

¹This code is adapted from the applet found at <http://www.cs.rochester.edu/u/brown/PLivewire/home.html>