



## Medical Image Processing 67705

*Semester A, 2020-2021*

### Homework 2

#### *Structures segmentation in CT scans*

Prof. Leo Joskowicz

TA: Adi Szeskin

Due: 29.11.2020

#### Submission Guidelines

- You are required to write and submit your own solution.
- Late submission policy: for each day after the submission date, 3 points will be subtracted from the total exercise grade. You may not submit the exercise if you are more than a week late.
- Submit (through moodle) a single ZIP archive file named “[id]\_[username]\_Targil1.zip”.

The ZIP file should contain:

- Your entire code (only .py files)
- A report as a pdf document including:
  - Your name and id
  - A description of your solution’s design and results for each part. Add graphs, screen snapshots, tables as needed to make your point clear.
  - Indications of which libraries – recommended and new -- you used in your code
  - A description of every function/method you wrote with a table of the following format:

Function Name with short explanation		
Input with explanation		
Output with explanation		
Image 1	Image 2	Image 3

## **Introduction and Background**

Segmentation, is an essential task in medical image processing. In this exercise, you will familiarize yourselves with the working environment for CT scans. You will explore the use of thresholding for segmentation of the skeleton and the aorta.

In medical image processing, perhaps in contrast to other courses you may have taken before, it is often the case that we do not have a single correct answer. You should try to achieve the best segmentation possible, but keep in mind that there is always observer variability and therefore you should not expect a perfect match (Dice coefficient of 1) with the published solution.

Solutions that are correct and within the estimated variability will earn you a grade of up to 90. The remaining 10 points will be given for originality and resourcefulness.

For this exercise, you are provided with 5 CT scans.

The CT scan files can be downloaded as

for  $1 \leq i \leq 5$ :

Case<i>\_CT.nii.gz

Case<i>\_L1.nii.gz

Case<i>\_Aorta.nii.gz

From the directory:

<https://drive.google.com/drive/folders/0B5wahAf6ThIuX2MxVnhYSW9faUk>

## Scientific Background

For exercise, we are interested in segmenting the body bones and the aorta. Here are the definitions and drawings of these structures.

- Bone (Wikipedia):

*A bone is a rigid organ that constitutes part of the vertebrate skeleton in animals. Bones protect the various organs of the body, produce red and white blood cells, store minerals, provide structure and support for the body, and enable mobility. Bones come in a variety of shapes and sizes and have a complex internal and external structure. They are lightweight yet strong and hard, and serve multiple functions.*

- Spine (Wikipedia):

*The vertebral column, also known as the backbone or spine, is part of the axial skeleton. The vertebral column is the defining characteristic of a vertebrate in which the notochord (a flexible rod of uniform composition) found in all chordates has been replaced by a segmented series of bone: vertebrae separated by intervertebral discs. The vertebral column houses the spinal canal, a cavity that encloses and protects the spinal cord.*

In this exercise we will look at a single vertebra called L1 (emphasized in the figure).

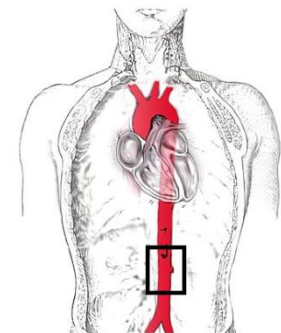
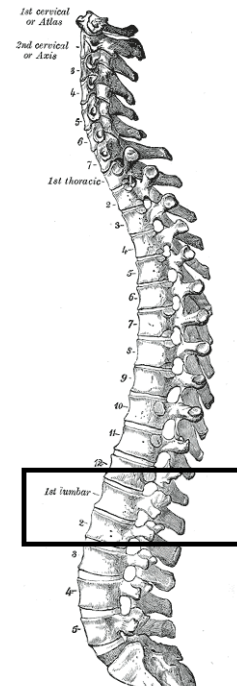
- The aorta (אבי העורקים – Wikipedia):

*The aorta is the main and largest artery in the human body, originating from the left ventricle of the heart and extending down to the abdomen, where it splits into two smaller arteries (the common iliac arteries). The aorta distributes oxygenated blood to all parts of the body through the systemic circulation.*

- CT scan, from Wikipedia:

*A CT scan or computed tomography scan (formerly computerized axial tomography scan or CAT scan) makes use of computer-processed combinations of many X-ray measurements taken from different angles to produce cross-sectional (tomographic) images (virtual "slices") of specific areas of a scanned object, allowing the user to see inside the object without cutting.*

CT scans are stacks of 2D grayscale images. The gray values in the images correspond to Hounsfield units. Read [here](#) to learn about CT and the typical HU values of various tissues.



## Technical Background

The standard file format for medical image is DICOM. In this exercise, we will use the NIFTI file format which is simpler and better suited for image processing. NIFTI files have the suffix is .nii. Compressed NIFTI files are .nii.gz.

NIFTI files can also be used to store segmentation masks. The information stored in a NIFTI file can be:

- Grayscale values: each voxel is an int16 data type, representing grayscale where black is the minimum value and white is the maximum value.
- Segmentation mask values: each voxel holds a value representing to which group it belongs. For instance: 0 – healthy tissue, 1 – old lesions, 2 – new lesions.

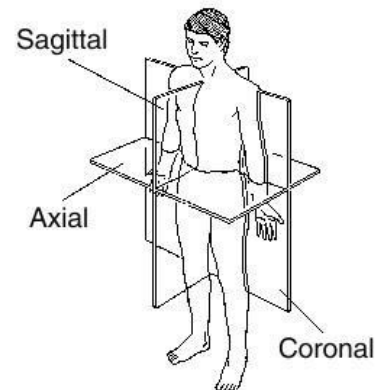
### Reading and writing NIFTI files

First, download and import the [nibabel library](#).

Then, you can read a 3D image using the get\_data method:

```
img = nib.load('Case1_CT.nii.gz')  
img_data = img.get_fdata()
```

Now, img\_data is a 3D image, with the first axis is the Sagittal plane, the second is the Coronal, and the third is the Axial.



### Viewing 3D images

Download and install the free [itksnap program](#).

This program allows for the display and interactive segmentation of 3D images. We will use it only for display. To do so, drag the grayscale image and choose 'Load as main image' and then drag the segmentation image and choose 'load as segmentation'. Note that this is possible only if both images share the same dimensions.

## **Part 1: Bones Segmentation in Contrast CT**

In this part you will perform threshold segmentation of the skeleton in a contrast CT (CT using radiocontrast).

For this purpose, you need to implement two functions – SegmentationByTH, which performs the segmentation using its input *Imin* and *Imax* thresholds, and SkeletonTHFinder which is used to find the best suited thresholds.

### **SegmentationByTH(nifty\_file, Imin, Imax):**

This function is given as inputs a grayscale NIFTI file (.nii.gz) and two integers – the minimal and maximal thresholds. The function generates a segmentation NIFTI file of the same dimensions, with a binary segmentation – 1 for voxels between *Imin* and *Imax*, 0 otherwise. This segmentation NIFTI file should be saved under the name <nifty\_file>\_seg\_<Imin>\_<Imax>.nii.gz.

The function returns 1 if successful, 0 otherwise. Preferably, raise descriptive errors when returning 0.

### **How to find *Imin* and *Imax*?**

Recall that bones have a maximal HU value of 1000+, so 1300 will do as the *Imax* for the segmentation.

The value of *Imin* is a bit trickier. You'll need to search for an optimal one:

### **SkeletonTHFinder(nifty\_file)**

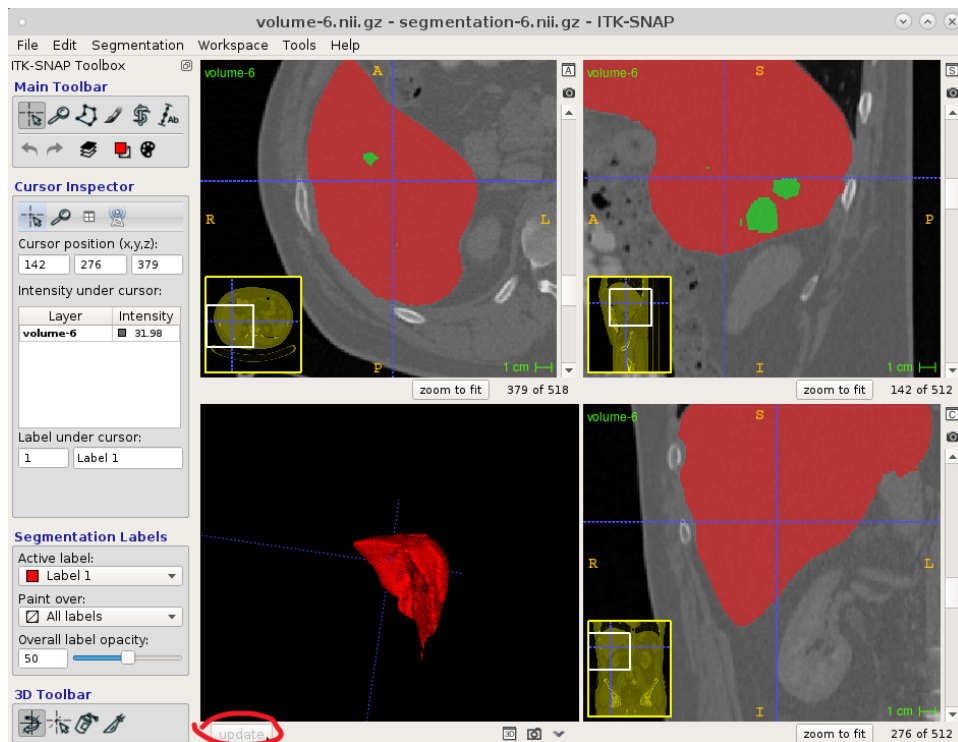
This function iterates over 25 candidate *Imin* thresholds in the range of [150,500] (with intervals of 14). In each run, use the SegmentationByTH function you've implemented, and count the number of connectivity components in the resulting segmentation with the current *Imin*. Plot your results – number of connectivity components per *Imin*. Choose the *Imin* which is the first or second minima in the plot. Also, make sure to include that graph in your report.

Next, you need to perform post-processing (morphological operations – clean out single pixels, close holes, etc.) until you are left with a single connectivity component.

Finally, this function should save a segmentation NIFTI file called "<nifty\_file>\_SkeletonSegmentation.nii.gz" and return the *Imin* used for that.

In your report, mention the final thresholds you used for each scan, and include snapshots of your result segmentations from itksnap.

Note that viewing the 3D segmentation in itksnap is computed by pressing the 'update' button:



An example of how your skeleton segmentation should look like (snapshot from itksnap):



## **Part 2: Aorta Segmentation in Contrast CT**

In this part you will perform segmentation of the aorta. Implement a function called `AortaSegmentation`:

**AortaSegmentation(nifty file, L1\_seg nifti file):**

This function is given a grayscale NIFTI file and an L1 segmentation NIFTI file. Use the L1 segmentation to tell on which axial slices you are required to segment the aorta. You may work on 2D axial slices to perform the segmentation here, and assume we are looking for a circle in each slice.

As for the algorithm in this part, develop it yourself using ROI, thresholds and morphological operations. Elaborate on your algorithm in your report.

For this part of the exercise, you are provided with Ground Truth segmentation NIFTI files of the aorta. To compare your results with the GT (whenever it is supplied), implement the function:

**evaluateSegmentation(GT\_seg, est\_seg):**

This function is given two segmentations, a GT one and an estimated one, and returns a tuple of (VOD\_result, DICE\_result). Use the definitions from the lecture slides.

Show your results in your report.

## **Appendixes**

Morphological operations:

<http://scikit-image.org/docs/dev/api/skimage.morphology.html>

Other useful functions:

<http://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.regionprops>

<http://scikit-image.org/docs/dev/api/skimage.measure.html#skimage.measure.label>