# ECE 4250 Final Report

Eldor Bekpulatov [eb654]

May 15, 2020

## Overview

Multi-atlas segmentation is an effective approach for automatically labeling objects of interest in biomedical images. In this final project, we were asked to manipulate multiple expert-segmented magnetic resonance (MR) images, called atlases, in order to maximally register each image to a target image and find their optimal deformation parameters. Next, the deformation parameters are then applied to the segmented data and the deformed atlas segmentations are combined using label fusion in order to determine the regions of interest (ROI) in novel images. This pipeline generates fast, reliable and consistent parcellations in novel MR images which can be highly accurate based on implementation. This report will focus on a general approach to solving this problem, then my effort to refine the generic pipeline in order to achieve better results.

## Baseline Design

In the first and the second milestones of this project, objective was to produce something that crudely can register each training case $(I_n, S_n)\ for\ n = 1, ..., N$ where $N$ is the number of training cases, to a target image $I_T$. Here, $I_n$ stands for MR image and $S_n$ stands for segmentation of the same image. Image and the segmentation are similar in shape, where a voxel or a point $p$ in $S_n$ directly corresponds to a point in $I_n$. First milestone focused on extracting the middle coronal slices of the MR images and the corresponding segmentations from each training and validation cases. Second milestone focused on deforming $I_n$ using 4 parameters $\{gs, rot, tx, ty\}$ to minimize the image differences between $I_T$ and $I_n$, where the difference indicator is a sum of squared difference (SSD) function. The 4 parameters specified above allow for a rigid body transformations such as global scale ($gs$), rotation ($rot$), and translation along x- and y-axis ($tx,\ ty$). Since the minimization modules did not have to be developed and using scipy's optimization module was permitted, the determining the optimization method to be used in the final version was an iterative plug-and-play process. With some parameter tweaks and min-max scaling the input images, Powell's minimization method, `fmin_powell`, was determined to be the best suited for this task, confirmed by visual inspections and relatively intuitieve parameter values. For the label fusion method, majority voting was selected, which simply counts the votes for each label from each warped atlas and chooses the label receiving the most votes to produce the final segmentation $S_T$.

## Alternative Design

Alternative design was built very much on top of the baseline design, and followed the same pipeline structure. Having laid a versatile code structure in baseline design allowed for a modular code base, where different components in the pipeline could be swapped in and out without breaking the pipe. Two main parts were implemented differently in the alternative design, transformation and label fusion module. Transformation module was swapped out to a 6-parameter one, instead of 4, $\{sx,\ sy,\ sh,\ rot,\ tx,\ ty\}$. This allowed for a greater flexibility, precisely an affine transformation, where one can control the scaling in the x- and y-axis independently ($sx,\ sy$). Also, an additional *shearing* ($sh$) parameter was added for another degree of freedom in our transformations. Optimization functions were also swapped out to inspect if the previously chosen method would result in a different outcome, however it was quickly realized that Powell's optimization method was still the best choice, although Nelder-Mead's method had significant improvements with the newer design.

Second most significantly modified module was the label fusion module. Here I had the most flexibility to work on and improve the registration results. It was realized that errors produced in atlas-based segmentation are mainly due to registration errors, i.e. registration associates wrong regions from an atlas to the target image. Several methods of label fusion were applied: *majority voting*, *weighted fusion*, and also *jointly weighted fusion* approach.

Let's start by modeling each method using mathematical notations and build the intuition to what I was able to achieve as the best registration result. *Majority voting* method is the most intuitive approach. It can be modeled as such:

$$S_T(x,y) \;=\; argmax_{l \in \{1 \ldots L\}} \sum_{n=1}^{N} S_n^l(x,y)$$

where $l$ indexes through labels and L is the number of all possible labels, $(x,y)$ indexes through all voxels of the image slice. $S_n^l$ is the vote for label $l$ produced by the $n^{th}$ atlas, defined by:

$$S_n^l(x,y) = \left\{ \frac{1 \; if \; S_n(x,y) = l}{0 \; otherwise} \right\}$$

Majority voting is very powerful in removing independent noise. When segmentation errors produced by different atlases are independent, the probability that multiple atlases agree on the same wrong label is exponentially suppressed compared to the probability that they agree on the same correct label. Hence, the combined results are expected to produce significantly fewer errors than those produced by any single atlas. Since majority voting assigns equal weights to different atlases, it makes a strong assumption that different atlases produce equally accurate segmentations for the target image. Thus, leading to my next approach.

*Weighted fusion* approach is where the training subjects (atlases) are weighted differently based on similarity between intensity values of $I_T$ and $I_n$. Initially, I simply calculated and sorted the SSD of the training ($I_n$) and the target ($I_T$) images and used the segmentation values directly from $S_n$. This approach can be modeled as such: $S_T(x,y) = S_n(x,y)$ where $n = argmin \sum_{i=1}^{N} SSD(I_T,\ I_i)$. However, this was not a suitable approach since all of the source of errors would be from the same atlas. To solve this, I had to introduce a convention of neighborhood $\mathrm{N}(x,y,r)$, which is a square region define by $(2r+1)\times(2r+1)$ at voxel $(x,y)$. This allowed for distribution of weights among different atlases using the SSD values bounded by $\mathrm{N}(x,y,r)$. Weighted fusion can be modeled by:

$$S_T^l(x,y) = \sum_{n=1}^{N} \omega_n(x,y) S_n^l(x,y)$$

where $\omega_n(x,y)$ is a local weight assigned to the $n^{th}$ atlas, with a constraint $\sum_{n=1}^{N} \omega_n(x,y) = 1$. I chose the weight distribution function can be modeled as such:

$$\omega_n(x,y) = \frac{1}{|\omega|} exp\{- \sum_{(x,y)\in \mathrm{N}(x,y,r)} SSD(I_T,\ I_n) \text{ where } |\omega| = \sum_{i=1}^{N} \omega_i(x,y).$$

Intuitively, greater the dissimilarity, which is modeled by $SSD(I_T,\ I_n)$, smaller the weight. After successive tests the optimal value of $r$ was determined to be 3. My best Kaggle submission was achieved using this method combined with min-max scaling intensity values at each voxel within neighborhoods defined by r = 3.

The last and final approach was to try jointly weighted label fusion, explained in a paper called *"Multi-Atlas Segmentation with Joint Label Fusion"*[1] . The overlaying idea in this approach is to create a square ( $N \times N$ ) dependency matrix $M_{x,y}(i,j)$ between each atlass for each voxel. $M_{x,y}(i,j)$ estimates how likely atlases $i$ and $j$ are to both produce wrong segmentations for the target image, given the observed feature images. Using this dependency matrix, the paper proposes to solve for $\omega(x,y)$ . The methods of solving for $\omega(x,y)$ and how to formulate the problem are provided in the paper, however the important takeaway is that this method eliminates the redundancy errors, meaning if two atlasses produce the same error, the probability of each producing an error is evenly split between the atlasses. This is extremely important because it eliminates the errors that can be caused by introduction of additional incorrectly labeled segmentations. Unfortunately, I spent a lot of time trying to implement the method proposed by the paper but to no avail. I spent a decent amount of time internalizing the paper and implementing a working version, I wanted to mention it in the final report. I personally would like to complete this implementation as a continuation to this class.

## Evaluation

The results achieved by my efforts were not exemplary. The registration algorithm that I developed produced relatively mediocre results. However, there was a linear progression of improvements observed in my efforts. Table 1 below outlines the succession of modifications made and the corresponding Kaggle scores achieved by each mode of operation.

**Table 1**. Modes of Execution and their Kaggle relative performance

| **MODE: constants={nearest neighbor interp, fmin_powell optimizer}** | **K Score** |
|---|---|
| 4-parameter rigid body, unscaled input, majority vote | 0.57678 |
| 4-parameter rigid body, min-max scaled input, majority vote | 0.57894 |
| 6-parameter affine, unscaled input, majority vote | 0.62749 |
| 6-parameter affine, min-max scaled input, majority vote | 0.66564 |
| 6-parameter affine, min-max scaled input, weighted fusion (r=20) | 0.66242 |
| 6-parameter affine, min-max scaled input, weighted fusion (r=10) | 0.66758 |
| 6-parameter affine, min-max scaled input, weighted fusion (r=5) | 0.68409 |
| 6-parameter affine, min-max scaled input, weighted fusion (r=3) | 0.68453 |
| 6-parameter affine, min-max scaled input, weighted fusion (r=2) | 0.68426 |

## Reflection

I thoroughly enjoyed the nature of the course; the final project was a good challenge and good representation of the types of engineering problems people face in different fields everyday. Luckily, there are a lot of brilliant minds pushing the boundaries of science and engineering to come up with novel solutions to challenges like these. It is very important that we, as a society, push for betterment of practices in fields like medicine, science, and engineering, for it is a vital part of our evolution as a species altogether. However, it is also crucial to realize the extent of the effects of our contributions to other people's lives. Having tools as such making the lives of medical professionals is very useful and possibly can make the lives of millions better. However, the same numbers that we evaluate our algorithms with can be the same numbers that can harm those whom we wish to help. I strongly believe that it is important that we stay true to our original goal, the betterment of our lives through pushing boundaries, and not be the cause of harm through carelessness. Particularly in medicine, I am a big proponent of regulated practices and upholding the highest of standards. I hold a cautious optimism about advances in technology.

## References

[1] Wang H, Suh JW, Das SR, Pluta JB, Craige C, Yushkevich PA. Multi-Atlas Segmentation with Joint Label Fusion. *IEEE Trans Pattern Anal Mach Intell*. 2013;35(3):611-623. doi:10.1109/TPAMI.2012.143

[2]Abraham, Chandler. "Computing 2D Affine Transformations Using Only Matrix Multiplication." *Medium*, Color and Imaging, 12 Jan. 2018