Implementation of the Functional Shape Cortical Brain Surface Analysis Pipeline on a High Performance Computing Environment

by

Yat Chung James Un

Project Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Engineering

in the
School of Engineering Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Summer 2016

Approval

Name:	**Yat Chung James Un**
Degree:	**Master of Engineering**
Title:	***Implementation of the Functional Shape Cortical Brain Surface Analysis Pipeline on a High Performance Computing Environment***
Examining Committee:	**Chair:** Ash M. Parameswaram Professor **Mirza Faisal Beg** Senior Supervisor Professor **Karteek Popuri** Supervisor Adjunct Professor
Date Defended/Approved:	August 8, 2016

Abstract

This project focuses on utilizing a surface registration method on multiple brain surfaces to obtain various geometric transformations for brain studies, and also the implementation of the analysis pipeline on a high performance computing (HPC) environment. Due to the infeasibility on runtime for performing surface registration between one template brain surface and multiple brain surfaces, an approach to perform sub-surface extraction on each brain surface and computation on a HPC environment has been employed. This has allowed a significant reduction in runtime and has also allowed the results to be obtained within reasonable time.

Table of Contents

List of Figures

Introduction

The Medical Image Analysis Lab (MIAL) was established by Dr. Mirza Faisal Beg and Dr. Ghassan Hamarneh in 2003. Research at the MIAL spans across mainly three core areas: segmentation and automated labelling of anatomical features, registration and geometric transformation of anatomical features, and statistical analysis via probability measures. Within these core areas are numerous projects for developing techniques that are aimed towards the overall theme of relating structural and functional aberrations to clinical disease within a precise anatomical framework. Some of these projects are: brain connectivity development from diffusion MRI, the isometric log-ratio transform for image analysis, shape denoising via template injection using binary Large Deformation Diffeomorphic Metric Mapping, etc. [1]

The main organ of the human central nervous system, the human brain, is one of the most complex organs in the human body. **Figure 1** provides an example of a human brain along with some of its regions and its cortical thickness data (distance between pial surface and white matter). For brain analysis, these data are often useful and necessary. The causes of many brain diseases and disorders, such as Alzheimer's disease, and their ideal treatments are still not fully understood with the current medical technology. With medical imaging techniques such as magnetic resonance imaging (MRI), magnetoencephalography (MEG), and electroencephalography (EEG), the data obtained from these techniques can be used to study and examine brain diseases and disorders. The Master of Engineering (MEng) project in this report utilizes surface registration on multiple brain surfaces from the medical data to obtain various geometric transformations for brain studies. In addition, this project also implements an analysis pipeline on a high performance computing (HPC) environment in an attempt to reduce the computation runtime.

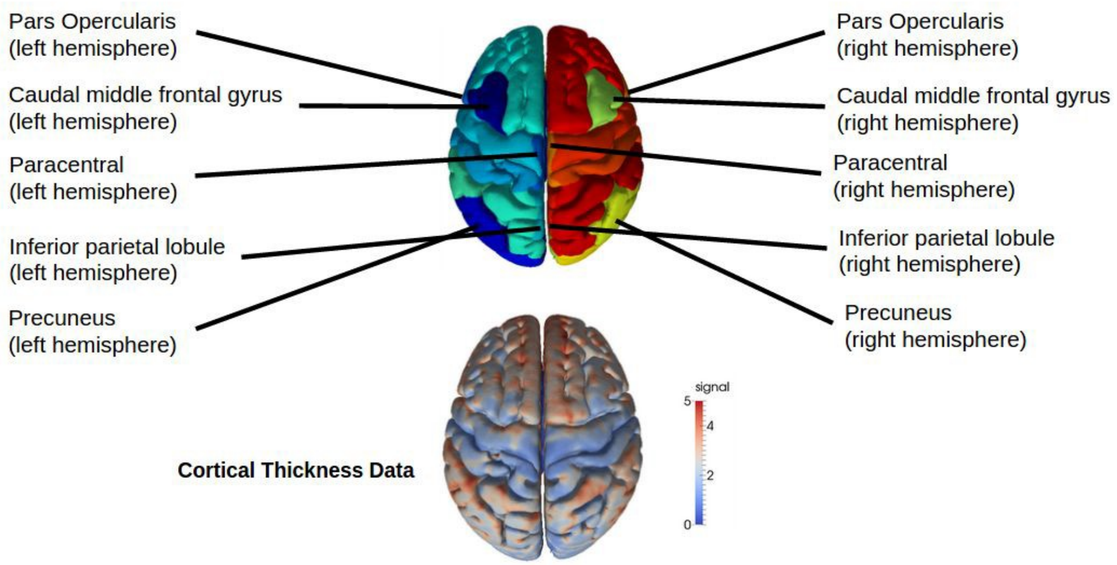

Figure 1. Regions of human brain and its cortical thickness data

Surface Registration

Surface registration is the process of aligning different sets of brain surface data, which are identified as targets, to one particular brain surface, which is the template. **Figure 2** demonstrates this by having one template transforming to several different targets. The targets and the templates can be misaligned due to different brain sizes and orientations, therefore it is necessary to first align them before making any comparisons. By realizing the transformation of one healthy brain to a brain with an unknown health condition, it is highly possible to identify any brain abnormalities if the transformation is vastly different from the typical transformations that resemble brains in good conditions. There are two categories of transformations in which surface registration algorithms can fall under. One is linear transformation, and the other is non-rigid/non-linear transformation. Linear transformation includes rotation, scaling, translation, and other affine transformations; while non-rigid transformation warps target brain surfaces to align with the reference brain surface.

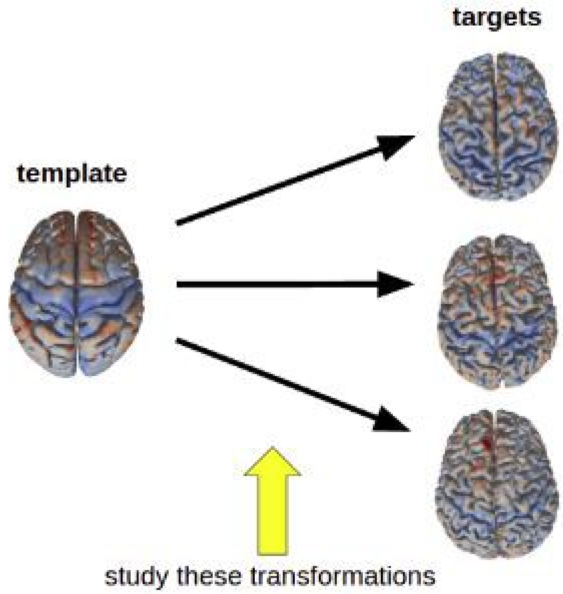

Figure 2. Surface registration

Method

Pipeline Development

The main concept of the analysis pipeline is to first perform linear transformation on the target brain surfaces to remove scale discrepancy, and correct for translational and rotational offset between the template and the targets. The next step would be to conduct non-rigid transformation, which is the fshape-registration in this case, to locally warp the template brain surfaces to match the target brain surfaces. The fshape-registration algorithm that performs non-rigid transformation is an important component of the analysis pipeline. The inputs of this algorithm are surfaces of targeted objects for registration, and these input surfaces should be structured in triangular meshes since they are discrete-data and cannot be completely identical to the actual objects. Small components, eyeballs for instance, have relatively small surface area, and as a result, they are suitable as inputs for the fshape-registration algorithm. However, the surface area of human brain surfaces are often too large for the algorithm. To understand this even further, it would be beneficial to compare the computation differences between linear transformation and non-rigid transformation. It takes 12 parameters to specify an affine transformation, which is the linear transformation in this pipeline; while it takes approximately 900,000 parameters for fshape-registration, which is the non-rigid transformation in this pipeline. As a result, fshape-registration is much more computation intensive than affine transformation.

To address the issue of oversized surface data and intensive computation during the fshape-registration process, there are two options to look into. First of all, it would be to decide if it is possible to extract the surface data of both template and target into multiple sub-surfaces, and then determine if registration can still be achieved with these sub-surfaces. The second option would be to abandon the current registration algorithm and search for a different one. Since the fshape-registration algorithm has already been used in the MIAL to a certain extent and it would create a huge change if the MIAL group would be using a different algorithm, therefore the other option should be considered as priority. Fortunately, it has been discovered that the current algorithm can perform non-linear registration with the extracted sub-surfaces, and the obtained results can still be comprehended by researchers.

Therefore, the main concept of the analysis pipeline is modified to conduct sub-surface extraction on the template and target data before undergoing fshape-registration, as shown in **Figure 3**.

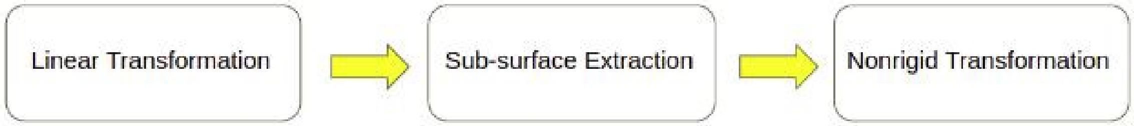

Figure 3. Steps for the analysis pipeline

The pipeline development can be considered complex since there are multiple computation tasks that need to be performed and it is necessary to interconnect the results from each of these tasks together. The computation time of each section of the pipeline is also a consideration, since computation resources are limited. In the process of registration, one template and one target are the minimum requirements. Since multiple targets are usually registered to the same template, it would be a good idea to perform registration on multiple targets at the same time. A simple method to provide a list of targets to the pipeline is to simply have a file that contains the target list as one of the pipeline inputs. This concept should reduce the workload of users and also reduce computation time, since users would not have to input the targets one at a time and the computation of one target is performed immediately after another one, instead of manually running the analysis pipeline for each target individually.

Due to the fact that only one template is used for multiple targets and the template surface is only required to be extracted into sub-surfaces once, therefore it would make sense to create a separate script to prepare for the template data. As a result, the analysis pipeline can be divided into the following steps:

1. Prepare template data by extracting the template sub-surfaces
2. Prepare target data by extracting the target sub-surfaces
3. Perform fshape-registration on each template sub-surface to corresponding target sub-surface

Step 1 - Prepare template data

The goal of this step is to prepare template data by extracting its brain surface into multiple sub-surfaces. In the MIAL, original template data is usually stored under a certain directory, and each template has its own identification name (ID). However, it would be useful to consider that template data can be non-standard and stored somewhere else. Therefore, the analysis pipeline is structured to allow users to have the option to specify if the template data is standard or not when deciding to extract template sub-surfaces. If it is a standard template, then only the template ID is required; otherwise, user is also supposed to provide the directory of the non-standard template data.

As for the process of extracting sub-surfaces, the regions of interest (ROI) should be defined for the purpose of obtaining the desired cortical regions from extraction. This is achieved by including the label file which belongs to the certain cortical surface file, since the label file maps and identifies different components of the cortical surface. Note that it is important to provide an accurate list of ROI in order to obtain the desired registration results, due to the fact that the final goal is to register the ROI of the template and the target.

Moreover, each surface file simply specifies the geometry of the brain surface, but information such as the cortical thickness is not included. Separate data files for each brain surface are used to define these functionality aspects, and these are identified as signals. During the extraction process, users can indicate which signal to include, so the program would not only extract the basic sub-surfaces but also the sub-surfaces with the indicated signal.

Last but not least, it would be a great idea to prevent unnecessary computation by checking if the selected template has already gone through extraction. This would preserve computation time and prevent overwriting existing files. If it turns out that not all the required files have been generated, then any existing files should be removed before restarting the extraction process. Once it has been determined that it is necessary to continue with the computation, the entire folder structure should be created following the checks. It is also a good convention to create all the folders in only one section of the script, since this allows the software development to be more organized.

Step 2 - Prepare target data

The step to prepare target data is similar to the step that prepares template data. The check for template data is also performed in this step, and extractions are performed on the targets based on the regions of interest. However, in order to properly register the template and the target, it is necessary to first align the target surface to the template surface using skull based affine alignment. This is the linear transformation process as explained earlier, and this process simplifies and reduces the workload of the fshape-registration process in the later step. Once the skull based affine alignment is complete, the target can then undergo the sub-surface extraction process which is identical to **Step 1**.

Step 3 - Registration

This step performs fshape-registration on each extracted template sub-surface to its corresponding target sub-surface.

Implementation Details

High Performance Computing Environment

After realizing that a vast amount of the pipeline computation is repetitive due to implementing the same steps on different targets, it has been decided that the pipeline can make use of distributed computing facilities such as computer clusters to improve its computation performance. A computer cluster consists of a set of connected computers that work together, and they can be viewed as a single system. This approach is particularly useful for the pipeline since there are multiple targets for registration. Computation for each target can be performed concurrently and individually on the cluster, and the computation time for all the targets can be approximately equivalent to that of one target.

Computation is performed using high performance computing (HPC) resources from the WestGrid organization. WestGrid provides Advanced Research Computing services for Canadian researchers [2], and it has computing facilities that offer special nodes with graphics processing units which are required for performing fshape-registration. **Figure 4** is the file structure for the analysis pipeline that would be suitable for sending jobs to the clusters. The cmd is a text file for users to provide pipeline inputs; while CERAMICCA is the Cloud Engine Resource for Accelerated Medical Image Computing for Clinical Applications, which is basically a website that would accept inputs as well.

The web_cmd basically sets up files and folders. The next step would be the wrapper and runner that handles the pre-processing procedure, and the last step would be the wrapper and runner for the fshape-registration process. In this analysis pipeline, the responsibility of wrappers is to check job status and submit jobs to the clusters, and the runners generate and run files on remote clusters.

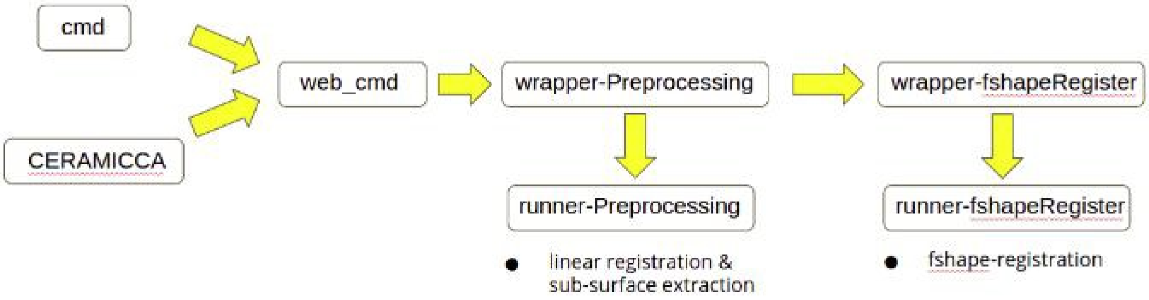

Figure 4. File structure

ROI Grouping

Realizing that after the sub-surface extraction step the computation for each sub-surface can be performed concurrently, a similar approach to the computation for each targets on a HPC environment can be used for the regions of interest (ROI). This should in theory further reduce computation time, with the trade-off being the number of jobs submitted to clusters increases. Each group would contain sub-surfaces such that each group would have approximately equivalent runtime, since this would have the least amount of runtime.

Scripting Languages and Programs

Bash scripting is the main scripting used for this pipeline. The reason for this is because the MIAL group has been using it as its main scripting as well, and it has powerful features such as reading and writing files, running programs, removing documents, etc.

MATLAB is another program that has been used. MATLAB is mainly used for constructing and running base scripts such as sub-surface extraction and fshape-registration for this pipeline.

The analysis pipeline has also been constructed with the framework of the MIAL scheduling system, which means that the API and certain basic functions have been provided and used for the creation of the pipeline.

Verification, revision and file versions

For every file and documentation modification, it is common practise to keep track of previous file version for times when comparison between previous and recent documents are required. Subversion (SVN) is the software used for this purpose. It is an open-source software, and members of the MIAL are able to oversee any changes made to files that have been committed to Subversion. Common commands for using Subversion are:

1. **svn commit** - this command recursively sends changes to the SVN server. It will commit changed files, added files, and deleted files.
2. **svn add** - this command lets the SVN server know that a file or directory has been created
3. **svn delete** - this command will delete the specified files from the local sandbox and the repository after committing
4. **svn update** - this command syncs the local sandbox with the server [3]

As for the analysis pipeline, it is important to verify if script modifications are producing the desired behaviors. In that case, the modified versions of the pipeline scripts need to be tested with sample data, and ideally these data should be closely related or even identical to the actual medical data, since this would highly resemble the actual use of the analysis pipeline. Furthermore, PARAVIEW can be used to visually verify the extracted sub-surface results.

Results

As mentioned earlier, sub-surfaces are split into multiple groups for the purpose of reducing computation runtime by increasing the number of jobs. A reasonable number of groups is 8, since the number of groups cannot be too much due to using shared HPC resources and therefore queuing time increases as the number of jobs, which in this case is proportional to the number groups, increases. That is also why jobs cannot be split by having one sub-surface for each group in an attempt to decrease runtime, because the increase of the queuing time will actually make the computation runtime increase. In addition, this strategy also prevents computation from exceeding the walltime limit of certain nodes on clusters, which would destroy the computed results if walltime limit is exceeded because the pipeline would identify this as a failed job.

The average runtime per target is approximately 17.5 hours, for both local computation and HPC. Depending on the number of targets, the local runtime is approximately 17.5 hours multiplied by the number of targets. The total HPC runtime remains as 17.5 hours, regardless of the number of targets due to embarrassingly parallel computation (little or no dependency). **Figure 5**, **Figure 6** and **Figure 7** demonstrates the result of affine alignment for one of the targets.

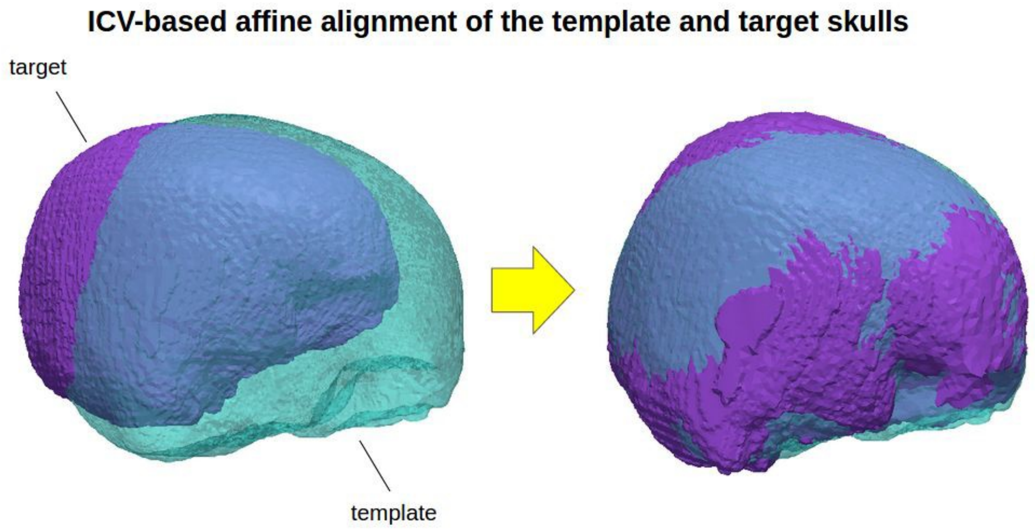
ICV-based affine alignment of the template and target skulls
target
template

Figure 5. **Skulls after affine alignment**

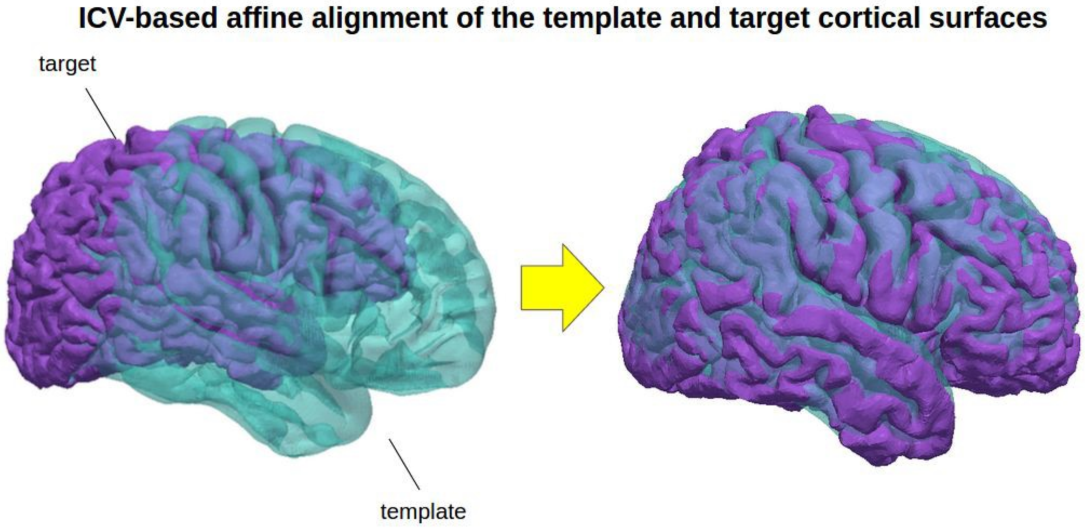
ICV-based affine alignment of the template and target cortical surfaces
target
template

Figure 6. Brain surfaces after affine alignment

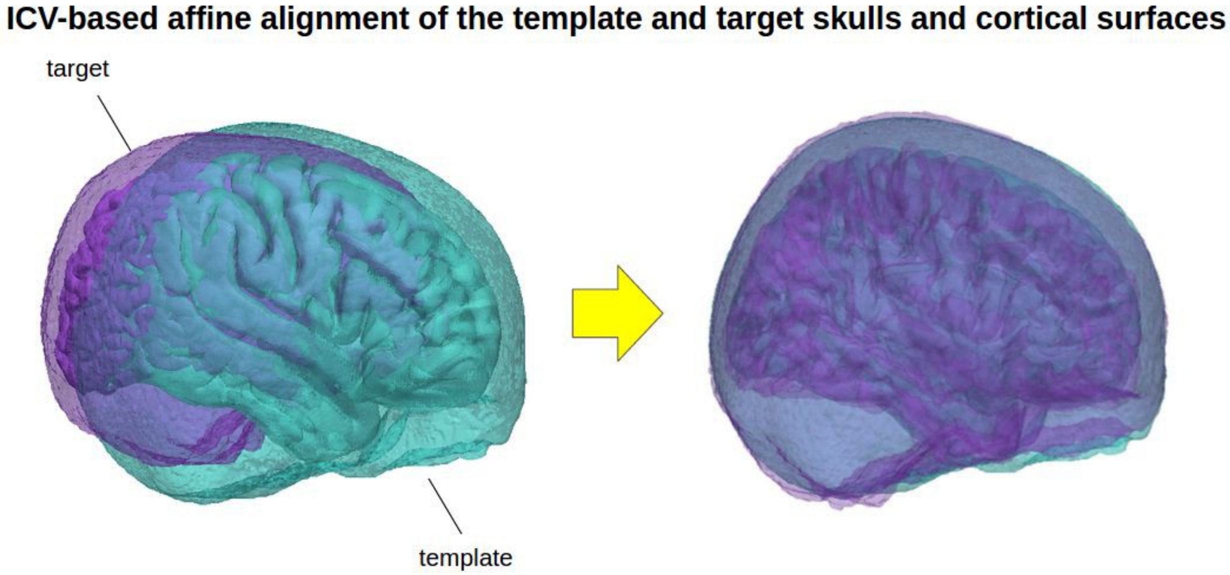
ICV-based affine alignment of the template and target skulls and cortical surfaces
target
template

Figure 7. Skulls and brain surfaces after affine alignment

One of the brain diseases that the MIAL group is interested in studying is the Alzheimer’s disease. Alzheimer’s is a type of dementia that causes problems with memory, thinking and behavior. Symptoms usually develop slowly and get worse over time. [4] One of the brain areas affected by Alzheimer’s is precuneus. **Figure 8** and **Figure 9** shows the result of the target precuneus after going through the analysis pipeline, which consists of the original template and target sub-surface, the sub-surfaces after affine alignment, the sub-surfaces after rigid alignment, and finally the sub-surfaces after fshape-registration.

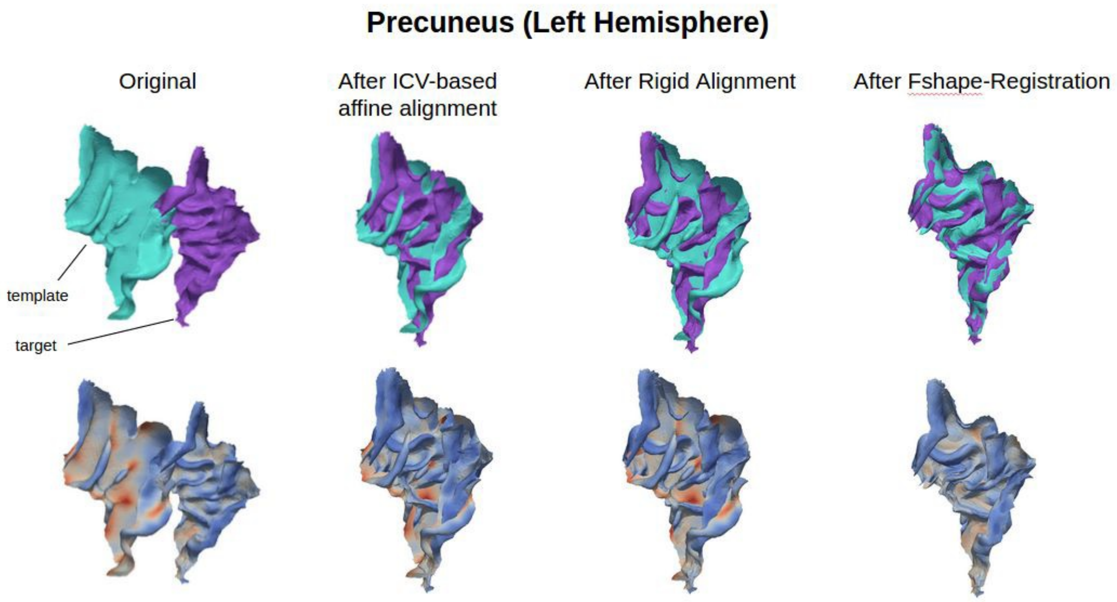
Precuneus (Left Hemisphere)
Original
After ICV-based affine alignment
After Rigid Alignment
After Fshape-Registration
template
target

Figure 8. Transformation on precuneus (left hemisphere)

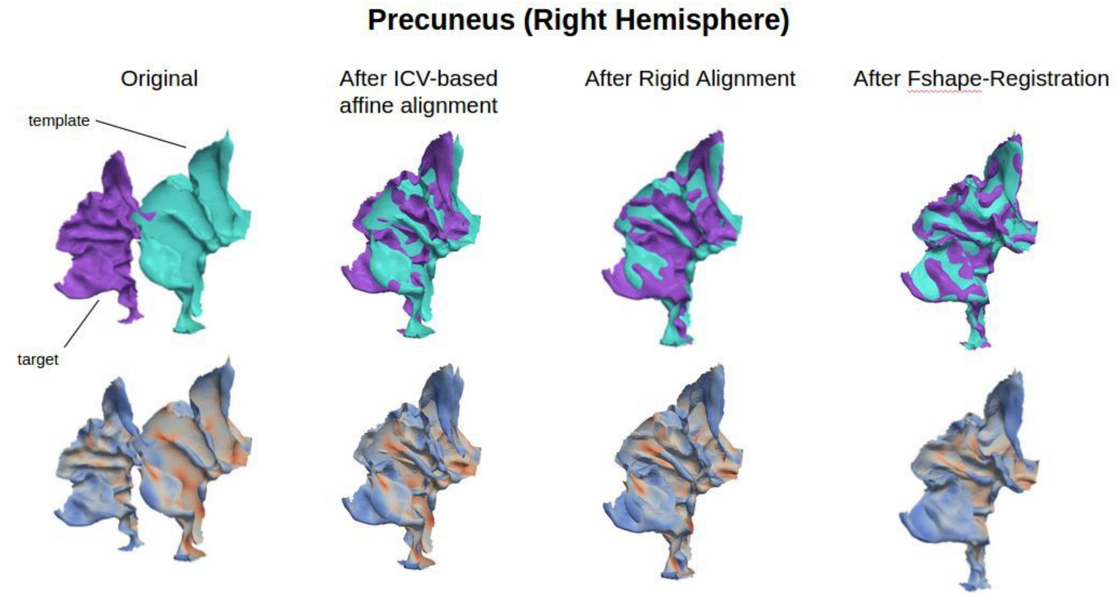
Precuneus (Right Hemisphere)
Original
After ICV-based affine alignment
After Rigid Alignment
After Fshape-Registration
template
target

Figure 9. Transformation on precuneus (right hemisphere)

Conclusion

The results from running the analysis pipeline on multiple targets have shown that the analysis pipeline has worked accordingly. **Figure 5**, **Figure 6** and **Figure 7** demonstrates that target brain surfaces are aligned with the template brain surface after affine alignment, and **Figure 8** and **Figure 9** shows the desired registered results for the precuneus component.

It has also been confirmed that the results from running the analysis pipeline on multiple targets in both the local environment and the HPC environment have produced predicted and desired outcome. If a local computation has been made, then the computation time is approximately 17.5 hours multiply by the number of targets, which can be a very unreasonable period of time if there are a large amount of targets. As for a computation in a HPC environment, the computation time remains as 17.5 hours, regardless of how many targets there are. The results have demonstrated that computation in the HPC environment significantly reduces the runtime from computation in the local environment if there are a large amount of targets, which in turn allow the results to be obtained within reasonable amount of time.

References

[1] Medical Image Analysis Lab. Simon Fraser University - Medical Image Analysis Lab. [Online]. Available: http://mial.sfu.ca

[2] Research Portal Home. Compute Canada. [Online]. Available: https://www.computecanada.ca/research-portal

[3] Chapter 3. Basic SVN Commands. Beyond Linux From Scratch Editor's Manual. [Online]. Available: http://www.linuxfromscratch.org/blfs/edguide/chapter03.html

[4] What is Alzheimer’s?. Alzheimer’s Association. [Online]. Available: http://www.alz.org/alzheimers_disease_what_is_alzheimers.asp