

Instructions (READ CAREFULLY)

- Complete individual steps and turn in your outputs (see **Task #**). There are a total of **6 Tasks** in this assignment.
- Paste (or screenshot) your corresponding codes to each task in your report. Your report should be self-contained. Please use a “pdf” format.
- Upload “pdf” (report) and “zip” (full source codes) to BlackBoard before the deadline.
- Feel free to discuss these problems with your classmates but turn in your own work; i.e., brainstorming together is encouraging but programming/writing should be done by yourself. **Turn in all the source code you write as well as detailed descriptions of your approach for the problems.**
- If you spend more than 24 hours on a single task, *very likely*, something goes wrong or you think too much. Ask the instructor for advice to save your time.

1. Data visualization

We will use hippocampal surfaces for this task. The subcortical structure is an important organ in the brain that is strongly associated with human memory. The shape is often obtained from structural T1-weighted images and has a relatively small volume to the cerebral cortex. In this project, hippocampal surfaces are already preprocessed by surface registration and re-tessellation. The surfaces thus have the same number of vertices ($\# = 1002$), where the vertex IDs are corresponding.

- **Task 1:** Let's visualize the input mesh files. Use ParaView and synchronize the viewpoint across all subjects. This is important step to see how the shapes are initially aligned. As studied, use “Link Camera” to sync all windows. Do not forget to enable parallel projection to avoid visual misinterpretation. Also, make the average shape by computing the mean 3D coordinates and visualize it with a separate plot.

2. Shape alignment

In the lecture, we studied the Procrustes alignment to find the best overlap. Once again, the surfaces are already re-tessellated after surface registration, but their coordinates are not updated (i.e., no shape alignment is performed yet).

- **Task 2-1:**
(Step1. initial alignment) Pick the first subject (1.vtk), and for each subject compute the best overlap to that subject by finding a rigid alignment (translation + rotation) via the Procrustes alignment. In MATLAB, you may want to use the following built-in function to update the vertex coordinates, which aligns v2 to v1.

`[~,v2new] = procrustes(v1,v2,'reflection',false,'scaling',false);`

Important note: reflection and scaling must be turned off to estimate a rigid alignment.

(Step 2. population average) Make the average shape by computing the mean 3D coordinates after the Procrustes alignment.

(Step 3. optimization) Repeat Step 1 with the average shape in Step 2 rather than the first subject (1.vtk).

You will keep estimating and updating the average shape after the Procrustes (i.e., compute average \rightarrow find Procrustes alignment to the average \rightarrow compute average \rightarrow find Procrustes alignment to the average \rightarrow ...). Let t denote # of iterations. We will repeat until the estimated average shape between $t-1$ and t is negligible, i.e., $\|S_t - S_{t-1}\|_F < 1e-12$, where $\|\cdot\|_F$ is a Frobenius norm.

- **Task 2-2:** Visualize your results in the same way you did in Task 1. Plus, plot the estimated average shape S_t , separately. You will observe significant difference between S_t and what you have in **Task 1**.

3. Principal Component Analysis

To understand shape variability, we will apply PCA on the aligned data in **Task 2**.

- **Task 3-1:** To run PCA on the aligned data, make a single column vector for each subject and concatenate them as a single matrix M . In MATLAB, you can easily compute PCA on your data. As the built-in function assumes row-wise data concatenation for PCA, you can call the function with the transpose of M as follows:

$$[pc, \sim, \lambda] = \text{pca}(M');$$

where pc is an ordered collection of principal components and λ is the associated eigenvalues (variances along the components). Compute PCA on the aligned data.

Hint: use `v(:)` to flatten out 3D coordinates into a single column vector.

- **Task 3-2:** We will explore the fraction of the total variance per component: i.e., the ratio between the eigenvalue of each component and the total variance. Rescale eigenvalues by the total variance (= a sum of all eigenvalues) to make sure a sum of the rescaled eigenvalues is equal to 1. Show a bar graph of λ to understand the amount of variation explained by each component. Also, compute the variation is explained by PCs 1 and 2; write down your answer.
- **Task 3-3:** Let's investigate 2 PCs associated with the largest eigenvalues. Visualize 7 plots per PC with the following configurations: $S_t - 3\sigma PC$, $S_t - 2\sigma PC$, $S_t - \sigma PC$, S_t , $S_t + \sigma PC$, $S_t + 2\sigma PC$, $S_t + 3\sigma PC$. You should have a single 2-by-7 plot. Here, PC is a 1002-by-3 matrix and σ is a square root of eigenvalues – do NOT use the rescaled eigenvalues here (use the original λ instead). Also, briefly describe how the mean shape changes along PCs in this dataset (e.g., shrinkage, bentness, inflation/deflation, etc.).

Hint: to convert a vector (PC, not a full matrix) to a matrix (coordinates), use “`reshape(pc, [], 3)`”.