Assignment 3 Report

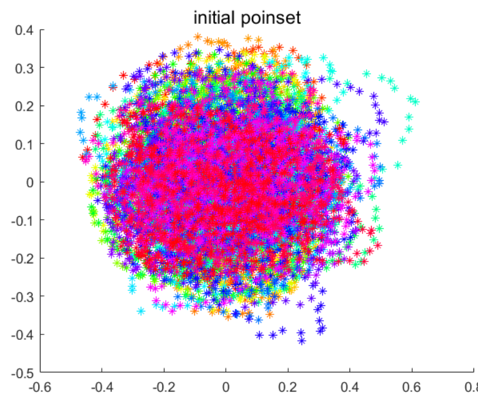
Jishnu Basavaraju, Akash Cherukuri, Silky Kumari

180050021, 190050009, 190070063

1 Shape Analysis on Simulated Shapes

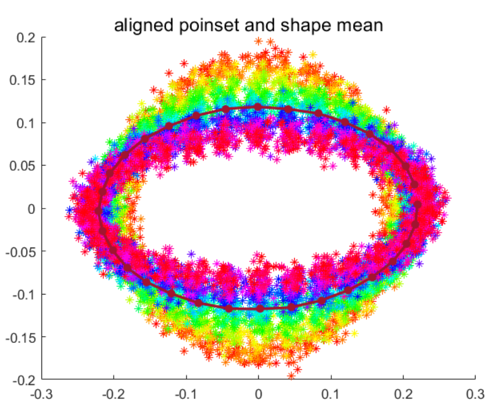
The submitted code contains all the required functionality implemented as a part of the shape analysis algorithm. This section contains the results as asked by the problem statement.

(d) The initial pointset is given below

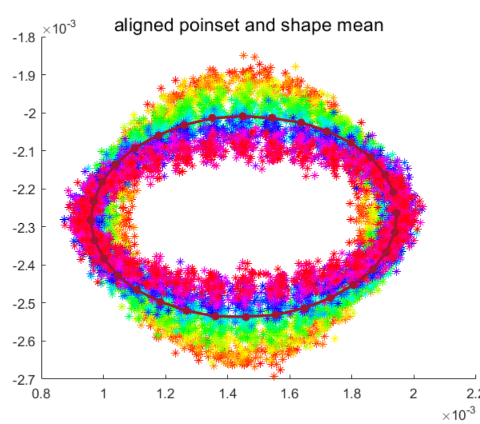


(e) The computed shape means from `code11` and `code22` are as follows:

`code11`

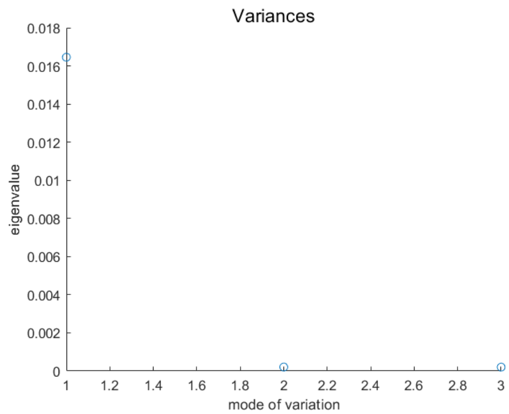


`code22`

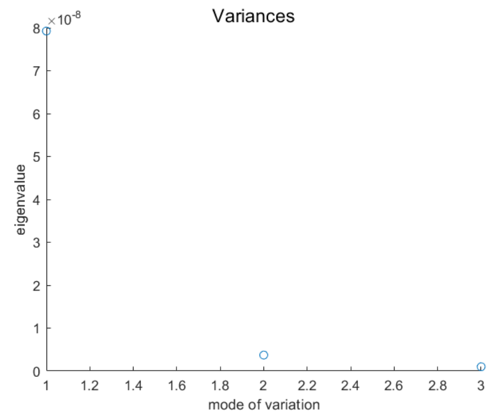


(f) Variance plots are as follows

`code11`

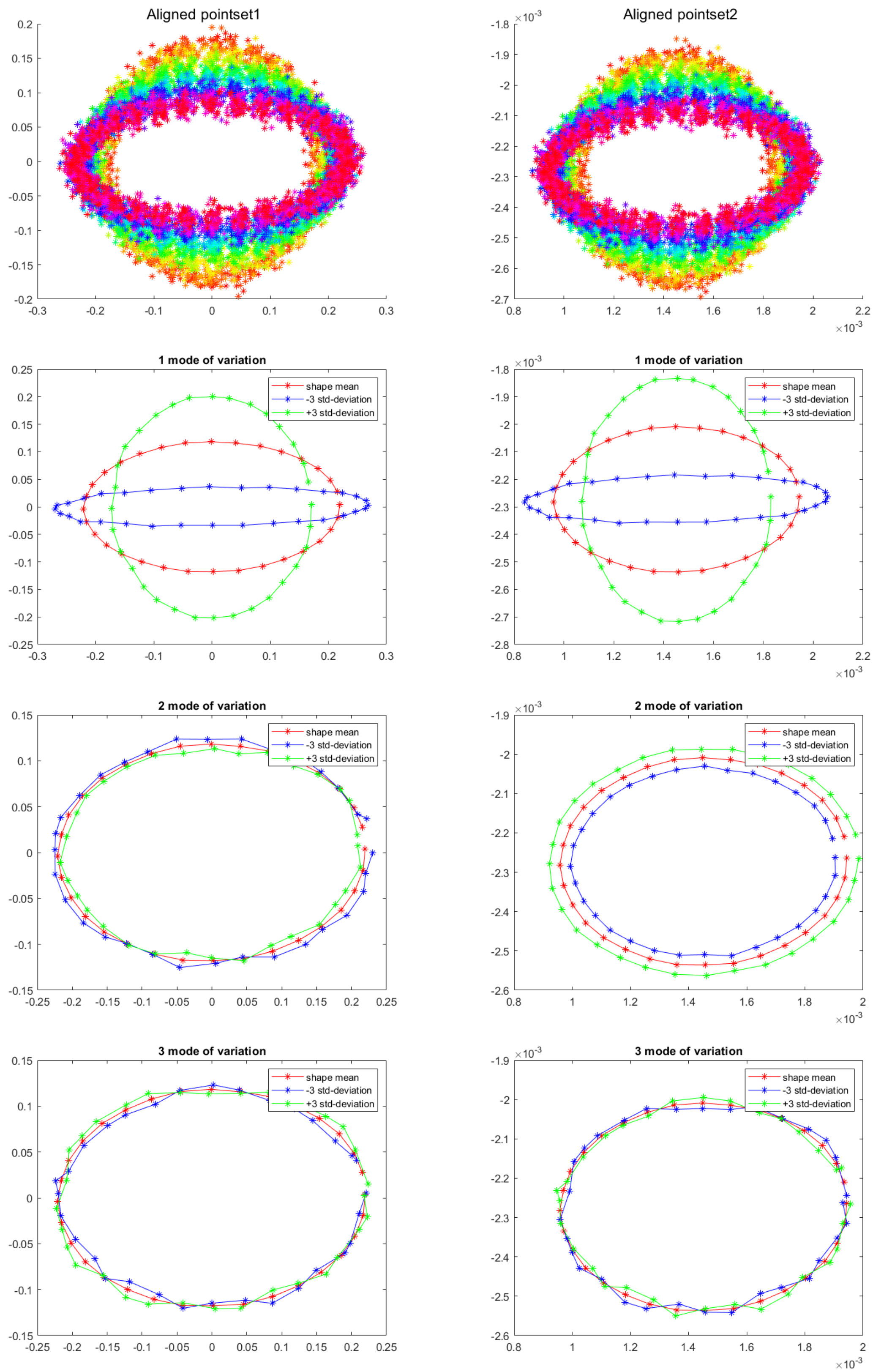


`code22`



(g) The plots showing the variation across the three principal modes is as follows:

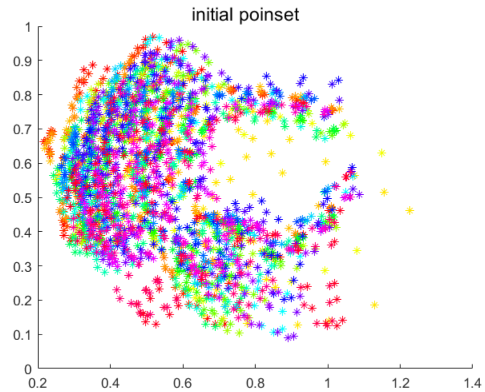
code11 code22



2 Shape Analysis on Human Hand Shapes

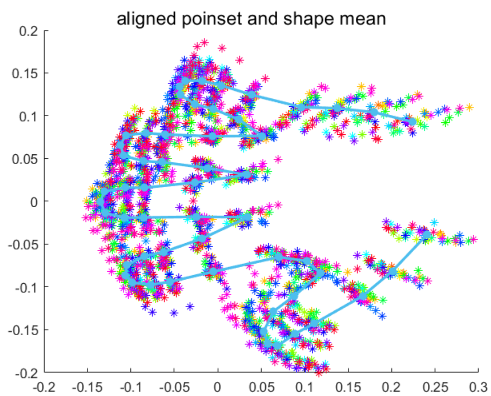
The submitted code contains all the required functionality implemented as a part of the shape analysis algorithm. This section contains the results as asked by the problem statement.

(d) The initial pointset is given below

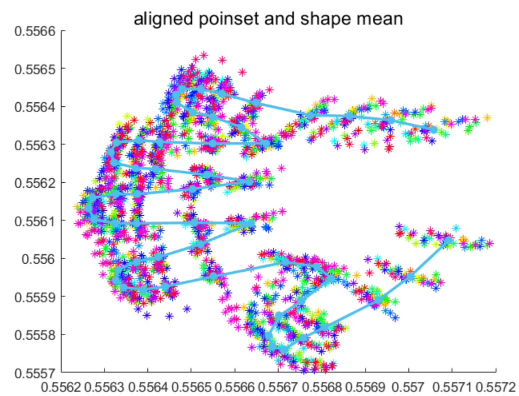


(e) The computed shape means from `code11` and `code22` are as follows:

`code11`

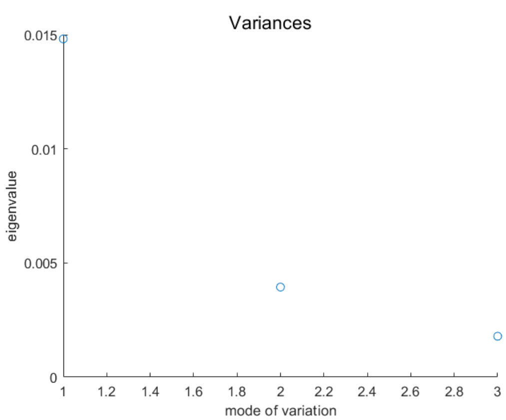


`code22`

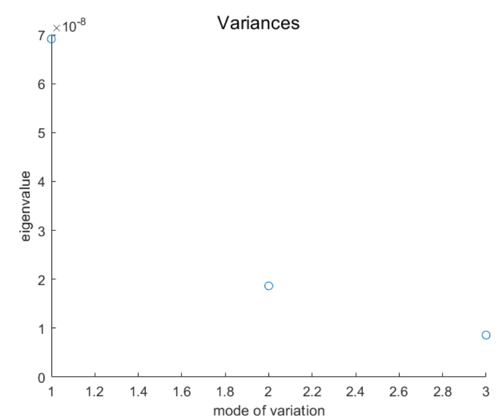


(f) Variance plots are as follows

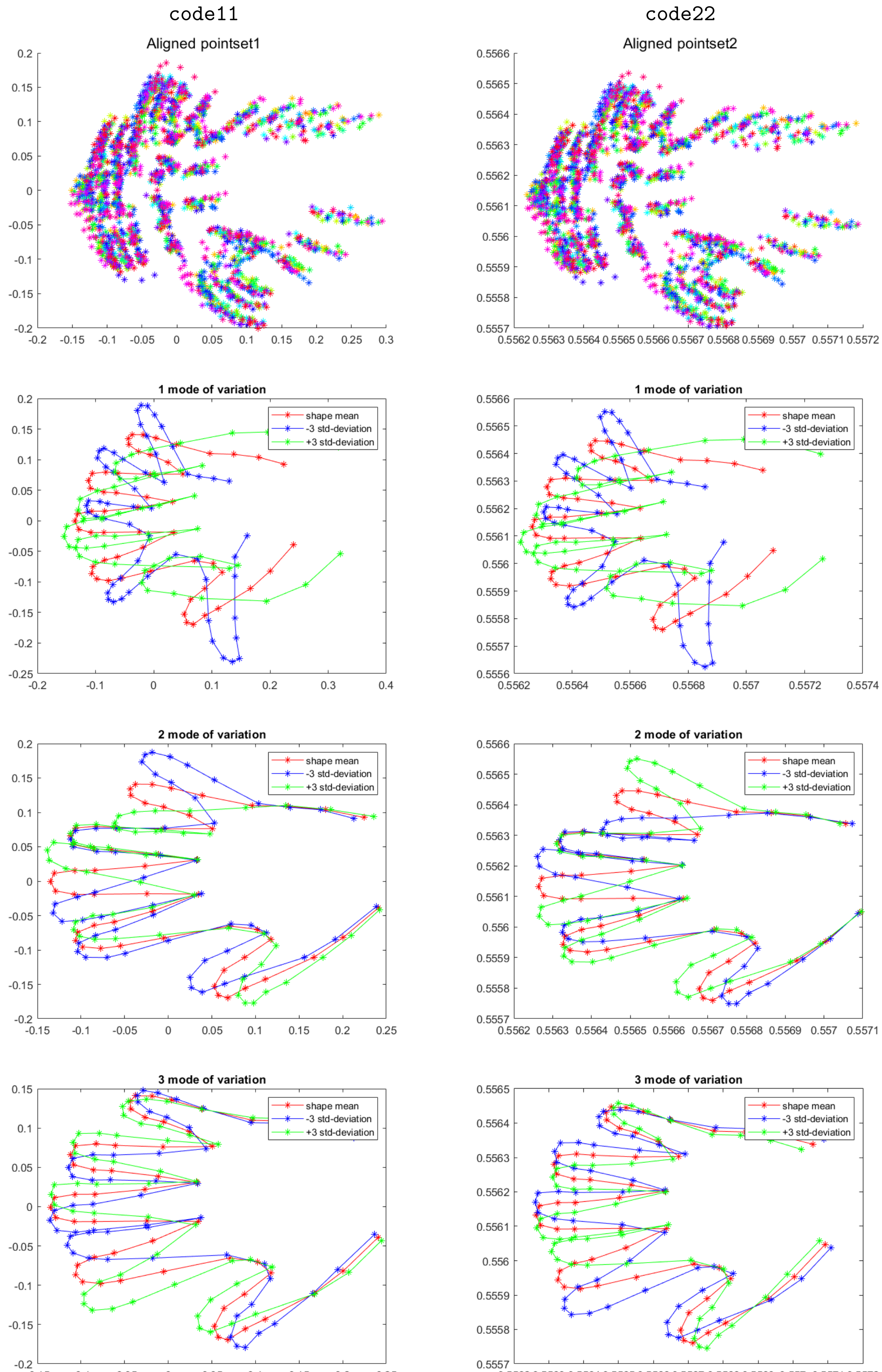
`code11`



`code22`



(g) The plots showing the variation across the three principal modes is as follows:



3 Clustering of Shapes

- We first define the notation used in the rest of the answer. z_n stands for the n^{th} point set, with cardinality N. Each point is D dimensional, with $D = 3$ in this case. The number of clusters is K.

  The Procrustes distance between two point sets z_1, z_2 is denoted as $\Delta(z_1, z_2)$. As we have already discussed in class, the mathematical definition for Procrustes distance is given by the following equation. S is the scaling matrix, M_θ is the rotation matrix and T is the translation matrix.

$$\Delta(z_1, z_2) = \min_{S,\theta,T} ||z_1 - SM_\theta z_2 - T||_2$$

- Let δ_{ij} be the identifier function for this clustering problem. It is defined as follows.

$$\delta_{ij} = \begin{cases} 1 & \text{if } i^{th} \text{ pointset is assigned to } j^{th} \text{ cluster} \\ 0 & \text{otherwise} \end{cases}$$

  The k^{th} cluster center is denoted as μ_k. The objective function to be minimized, J can now be defined as follows. $L_c(\mu_k)$ is a set function to take care of outliers that will be defined in the next section.

$$J = \sum_{j=1}^{K} \left[\sum_{z_i \in L_c(\mu_k)} \Delta(z_i, \mu_k) \right]$$

- The algorithm is quite similar to how the KMeans algorithm works. We assume that the means have unit norm, are centered at the origin and are aligned along a pre-determined hypersphere. These constraints allow the means to effectively convey the "shape" that they represent.

Initialization

A bottom up initialization akin to the Kmeans algorithm is performed. That is, we find the Procrustes distance between every pair of points, and at every iteration we "merge" the closest pair by replacing them with their mean. The "mean" shape can be found by using the algorithm discussed in class.

The initialization for μ_k is done when there are K clusters left, and all the points that were merged to form the k^{th} cluster are assigned to it.

$$\mu \textit{ and } \delta \textit{ have been initialized!}$$

Algorithm for Clustering

Akin to the KMeans algorithm, an alternating approach between optimizing the memberships and optimizing the class means has been chosen. One iteration of the optimization algorithm corresponds to both the steps being performed.

1. **Optimize Memberships keeping Class Means fixed**
   Every pointset z_i is assigned to the class with class mean μ_k iff the following equation holds.

$$k = \arg\min_{k'} \Delta(z_i, \mu_{k'})$$

   That is, each pointset is assigned to the closest class mean. It can be proven using the exchange argument that such an assignment is optimal.

2. **Optimize Class Means keeping Memberships fixed**
$L_c(\mu_k)$ is a subset of all the points that assigned to the k^{th} class. Let σ denote the standard deviation and M the mean of the Procrustes distance of all the points in the k^{th} class and μ_k. The set is mathematically defined as follows.

$$L_c(\mu_k) = \{z_i \ : \ \delta_{ik} = 1 \text{ and } \Delta(z_i, \mu_k) < M + c\sigma\}$$

That is, $L_c(\mu_k)$ does not contain the pointsets which lie far away from the cluster center which are very likely to be the outliers. Note that c is a constant that needs to be set by the user prior to the algorithm being run.
Therefore, the k^{th} class mean is updated to be the "mean shape" for all the points that belong to the set $L_c(\mu_k)$. Again, this mean shape can be computed using the algorithm given in class.

Stopping Criterion

The algorithm may terminate after a set number of iterations is done, or when the change in the loss function is smaller than a constant d given by the user.