The Pennsylvania State University
The Graduate School

# STATISTICAL SHAPE ANALYSIS USING DEFORMETRICA: ESTIMATING MEAN SHAPE AND GEODESIC SHAPE TRAJECTORY

A Thesis in
Statistics
by
Mithun Kumar Acharjee

© 2021 Mithun Kumar Acharjee

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2021

The thesis of Mithun Kumar Acharjee was reviewed and approved by the following:

Matthew Reimherr
Associate Professor of Statistics
Thesis Advisor

Jia Li
Professor of Statistics

Ephraim Hanks
Associate Professor of Statistics
Chair of Graduate Studies

# Abstract

Statistical shape analysis is an emerging field of research that analyzes the geometrical properties of a given set of shapes or objects using different statistical methods. Two important aspects of the shape analysis are to estimate the mean shape from a given set of shapes and to estimate a shape trajectory as close as to the observed shapes in order to determine the continuous evaluation of shapes over time. A deterministic Atlas model is used to compute the mean shape (also referred to as atlas construction) from a set of shapes which builds a generalization of a typical representation by preserving the characteristics of the original shapes. Thus mean shape is useful in forecasting trends in form or pulling out stereotypes from a set of homologous shapes. The Geodesic regression is used to estimate the continuous shape evaluation at a certain time within its intervals where the mean face obtained from the deterministic atlas model can be used as baseline shape or initial template face to initiate the program.

In this thesis, we are showing the application of the deterministic atlas model and geodesic regression model using a shape analysis software called Deformetrica. We collected data from the three Dimensional Facial Norm (3DFN) database which provides craniofacial anthropometric normative data deposited in the FaceBase consortium. Our data are 3D facial mesh where each facial mesh contains a high number of landmark points. We applied the deterministic atlas model using Deformetrica accessing the GPU allocation, which helps to estimate the mean facial object. We used this mean facial object as an initial template shape on geodesic regression which provides an estimated shape trajectory of the facial objects. This geodesic shape trajectory is a geodesic flow of diffeomorphisms acting on the above baseline template shape to estimate the continuous 3D facial evaluation with age varying continuously within its range. This thesis also describes the technical details of using Deformetrica in a high-performance computing environment while dealing a 3D geometric objects with a high number of landmark points.

# Table of Contents

**List of Figures** vi

**List of Tables** viii

**Acknowledgments** ix

**Chapter 1**

**Background** 1

1.1 Introduction . . . 1
1.2 Shape and Landmarks . . . 2
1.3 Shape Analysis Methods: . . . 6
1.3.1 Traditional Methods: . . . 6
1.3.2 Geometrical Methods: . . . 6
1.4 Shape Alignment . . . 7
1.4.1 Shape Space . . . 7
1.4.2 Procrustes Analysis . . . 8
1.4.2.1 Ordinary Procrustes Analysis (OPA) . . . 9
1.4.2.2 Generalized Orthogonal Procrustes Analysis (GPA) . . . 10
1.4.3 Different Shape Alignment Methods . . . 13
1.5 Theoretical Background: Deterministic Atlas . . . 14
1.5.1 Large Deformation Diffeomorphic Metric Mapping (LDDMM) . . . 14
1.5.1.1 Parametric Family of Deformation: Constructing Diffeomorphisms . . . 14
1.5.1.2 Distance between Shapes: Meshes or Images . . . 16
1.5.1.3 Loss function for the Deterministic Atlas Model . . . 17
1.6 Theoretical Background: Geodesic Regression . . . 18
1.7 Tools and Techniques: Atlas Construction . . . 19
1.7.1 Deformetrica Installation: . . . 20
1.7.2 Configuration File Set-up: . . . 20
1.7.3 Atlas Estimation: . . . 22
1.7.4 Output files . . . 22
1.7.5 Parallel Computing: CUDA Acceleration . . . 23
1.8 Tools and Techniques: Estimating Shape Trajectory . . . 23
1.8.1 Configuration file set-up . . . 23

1.8.2 Geodesic regression estimation . . . 24
1.8.3 Output files . . . 24
1.9 Benefits of the Study: . . . 25
1.10 Contribution of the Study: . . . 25

**Chapter 2**
**Results** **26**
2.1 Data . . . 26
2.2 Data Processing . . . 26
2.3 Application of Deformetrica: 3D Facial Norms Atlas Construction . . . 27
2.4 Application of Deformetrica: Estimating Geodesic Regression . . . 28

**Chapter 3**
**Conclusion** **32**
3.1 Discussion . . . 32
3.2 Limitation . . . 34

**Appendix**
**Computer Programs** **35**
1 XML Configuration Files . . . 35
1.1 Deterministic Atlas Model: Data XML File . . . 35
1.2 Geodesic Regression Model: Data XML File . . . 36
1.3 Deterministic Atlas Model: Model XML File . . . 37
1.4 Geodesic Regression Model: Model XML File . . . 37
1.5 Deterministic Atlas Model: Parameter Optimization XML File . . . 38
1.6 Geodesic Regression Model: Parameter Optimization XML File . . . 38
1.7 Deterministic Atlas model execution: . . . 39
1.8 Program running from Roar Super Computer . . . 39
2 MATLAB Code: Converting vtk file to obj file . . . 39
3 R Code: Converting obj file to vtk file . . . 41

**Bibliography** **43**

# List of Figures


1.1 Two outlines of the same second thoracic (T2) vertebrae of a mouse having different Euclidean similarity transformation [33] . . . . . 3

1.2 Four copies of the same shape having Euclidean similarity transformation [34] 3

1.3 Landmarks representing a shape [35] . . . . . 4

1.4 Anatomical landmarks pointed on a macaque monkey skull [36] . . . . . 4

1.5 6 mathematical landmarks (diamond round a +) and 42 pseudo-landmarks (+) located on T2 mouse vertebra [36] . . . . . 5

1.6 Geometrically transforming one species of fish, Diodon (the source), into another species, Orthagoriscus mola (the target) [41] . . . . . 7

1.7 Optimal Procrustes superimposition of two geometric shapes [62] . . . . . 8

1.8 Left: 40 unaligned shapes. Right: 40 aligned shapes with the mean given in red [64] . . . . . 13

1.9 Shape regression used four time indexed observations (solid surfaces) to estimate the continuous shape evaluation (transparent surfaces) [69] . . . 18

1.10 Estimated geodesic regression. Top row: the estimated trajectory. Bottom row: observations from which the top trajectory is learned. [31] . . . . . 19

2.1 Eleven cleaned 3D Mean Face based on 11 clusters each having 50 objects 29

2.2 A cleaned 3D Mean Face based on 550 facial norm shape objects (or, based on 11 cluster mean face objects) . . . . . 30

2.3 Estimated geodesic regression. First two rows: Estimated geodesic trajectory or continuous evaluation of 3D facial norm shape. Last row: Observed data from which the top trajectory is learned . . . . . 31

# List of Tables

2.1 Program execution time using GPU allocation . . . . . 30

2.2 Faces with corresponding ages . . . . . 30

# Acknowledgments

I would like to thank my Supervisor, Dr. Matthew Reimherr for his support and encouragement. I am grateful to him for his guidance and valuable advice throughout the study and preparation of the manuscript. I would like to acknowledge my gratitude to Dr. Jia Li to read the thesis and made some valuable comments to improve the research work. I would like to thank Mr. Omar Ibrahim Hagrass and Dr. Carlos J. Soto for their help in various steps of this research work. Special thanks go to my beloved wife Anamika Datta and family members for their continuous support throughout this research work.

# Chapter 1
# Background

## 1.1 Introduction

Statistical shape analysis is an emerging field of research and widely used in a wide variety of disciplines such as biology [1–8], image analysis [9–11, 82–85], computer vision [77–81], medicine [12, 13], geology [14, 15], archaeology [16, 17], chemistry [18–20] and genetics [21]. Statistical shape analysis uses statistical methods to analyze the geometric properties (e.g., position, orientation, and size) of a given set of objects. For instance, statistical shape analysis could be used to measure the size and shape difference between young and adult monkey (sooty mangabey-a type of monkey) skull [22]. Statistical shape analysis uses different shape statistics to quantify significant aspects of shape such as to estimate mean shapes from a given set of shapes, to estimate shape variability between shapes and shape deformation i.e., transforming one distinct shape to another one and to estimate a continuous shape evaluation from a discrete set of shapes at a specific time within the time intervals [23, 24].

Before the advent of the computer, D'Arcy Thomson, one of the pioneers of mathematical biology, first introduced the idea to transform one distinct shape into the other through the ambient-space deformations in his groundbreaking book "On Growth and Form" [25]. After so many years, the shape analysis concept still relevant where the vast improvement happened over the last 30 years. One such significant improvement is to develop Large Deformation Diffeomorphic Metric Mapping (LDDMM) [26–30], a concurrent and novel framework for the construction of such transformations.

Deforming 3D shapes or performing Procrustes analysis to find an average 3D object configuration is not an easy task. It requires a high-performing computer facility. In this study, we will discuss our experience regarding GPU-based computing of the Deterministic Atlas model using Deformetrica, a shape analysis software. Also, we

will use the mean face obtained from the Deterministic Atlas model to perform the Geodesic Regression (another important application of Deformetrica) which will provide the continuous shape evaluation over time. We choose Deformetrica for two specific reasons: First, it has very few data requirements. For instance, it does not require point correspondence between shapes/objects [31]. Second, it is an open-source shape analysis software (`http://www.deformetrica.org/`). The LDDMM framework is considered as the skeleton of the deterministic atlas model for constructing atlas/mean shape and geodesic regression for estimating a geodesic shape trajectory learned from the observed shape objects. Deformetrica relies on a specific instance of the LDDMM framework, based on control points [26, 31]. Before explaining the mathematical background of the Deterministic Atlas model [section 1.5] and its implementation using Deformetrica [section 1.6], it is necessary to understand the geometry behind shape objects, shape analysis methods, and shape alignment process.

## 1.2 Shape and Landmarks

Technological advancements give the opportunity to collect geometric information from different objects whether the object is man-made or from nature. Thus the study of the shape of the object is extremely important. In the beginning, it is important to know what is the shape? We adopted an intuitive definition provided by D.G. Kendall [32]:

**Definition 1.2.1.** *Shape* is all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object.

This definition implies that an object's shape is invariant under the isomorphic transformation of translation, rotation, and scaling. Note that this isomorphic transformation is also known as Euclidean similarity transformation. Figure 1.1 and Figure 1.2 shows the invariant property of shape by different Euclidean similarity transformation.

The next important question that strikes our mind: how we can describe a shape? From ancient times, people often prone to compare an unknown shape with a known form. For instance, the shape of a Pakistan on the map looks like a dog. But this subjective judgment is not related to the algorithm framework. A scientific way is to look into points on each object which is known as a landmark [Figure 1.3]. Dryden and Mardia [23] provided a suitable definition of the landmark:

**Definition 1.2.2.** A *landmark* is a point of correspondence on each object that matches between and within populations.

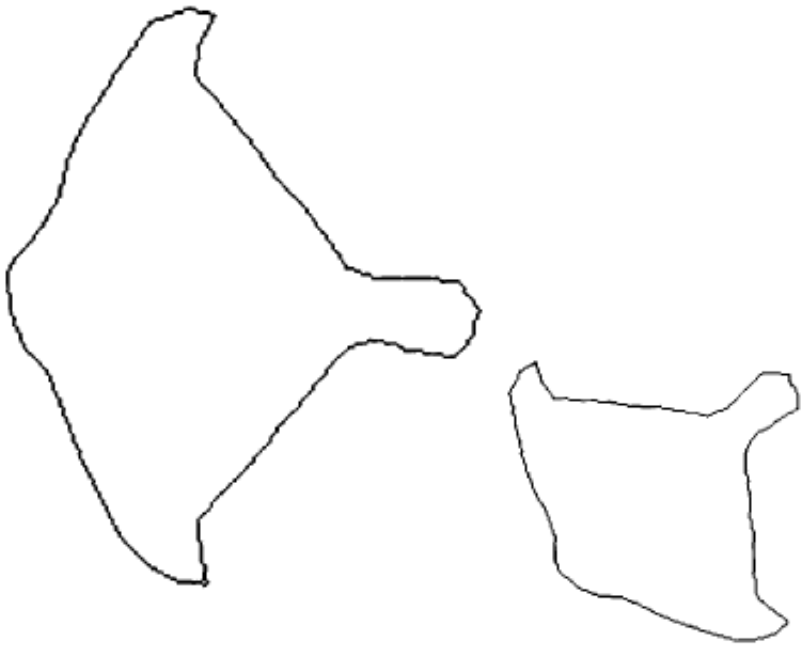

**Figure 1.1.** Two outlines of the same second thoracic (T2) vertebrae of a mouse having different Euclidean similarity transformation [33]

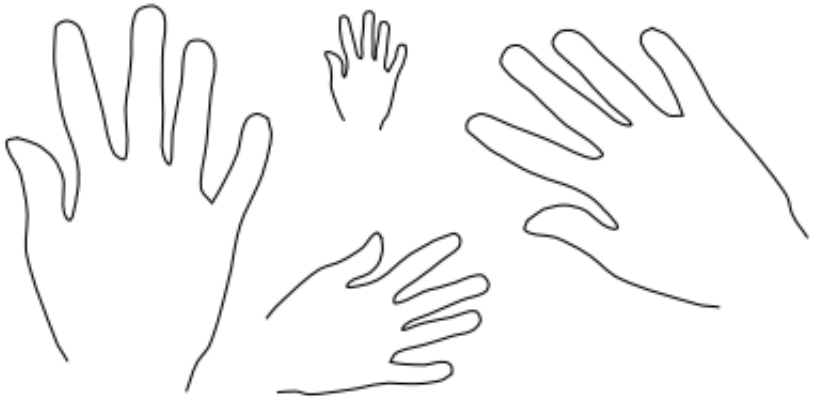

**Figure 1.2.** Four copies of the same shape having Euclidean similarity transformation [34]

Dryden and Mardia [36] differentiate three basic types of landmarks and provide definitions and related discussions. The three landmarks are anatomical, mathematical, and pseudo-landmarks.

Anatomic landmark is a point assigned by an expert say doctor to portray a biological object or object. An anatomical landmark can be a corner point of an eye. Figure 1.4

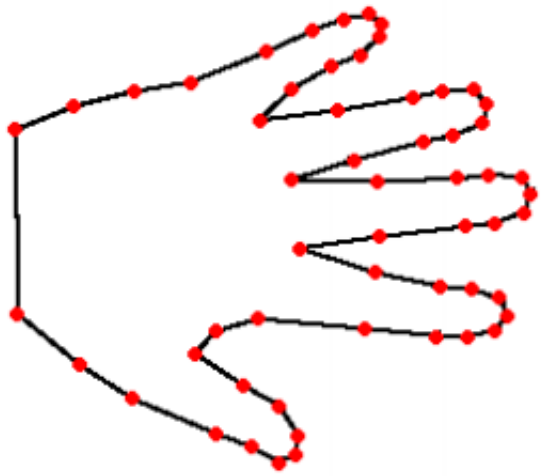

**Figure 1.3.** Landmarks representing a shape [35]

depicts some anatomical landmarks pointed on a macaque monkey skull.

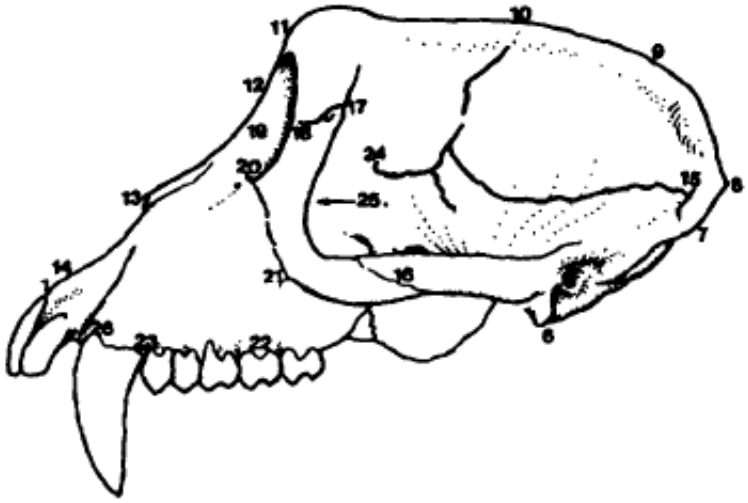


**Figure 1.4.** Anatomical landmarks pointed on a macaque monkey skull [36]

Mathematical landmark is a point assigned by some mathematical or geometrical characteristics. Mathematical landmarks could be a point at high curvature. In figure 1.5, the diamond shape plus points are at high curvature and represent the 6 mathematical landmarks. In an intention to identify automatic pattern recognition, a computer program

determine the mathematical landmarks.

Pseudo-landmarks are a constructed points that are assigned between the other types of landmarks (say, between two anatomical landmarks or between two mathematical landmarks) or assigns around the continuous curves or outlines. Figure 1.5 shows pseudo-landmarks (42 plus sign around the mouse vertebrae) assigned through the continuous curves or outlines between two mathematical landmarks. Some shape matching process requires a large number of points. Pseudo-landmarks are particularly helpful in that situation as it around the continuous curvature or outlines.

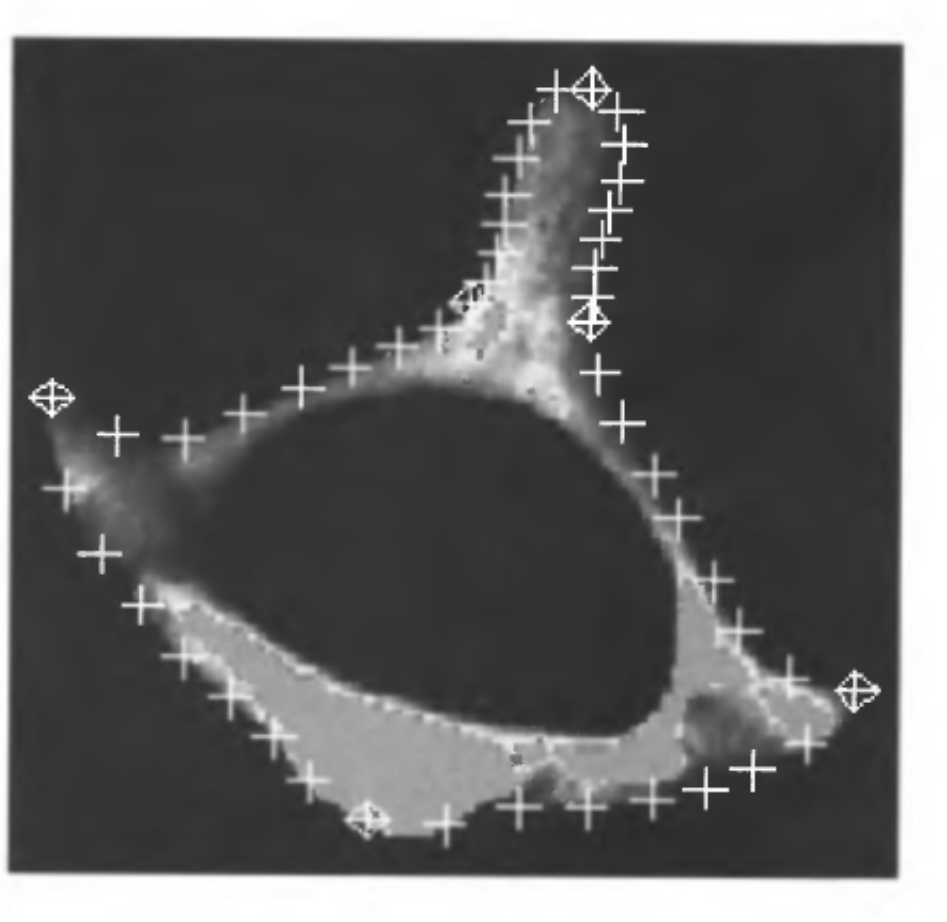

**Figure 1.5.** 6 mathematical landmarks (diamond round a +) and 42 pseudo-landmarks (+) located on T2 mouse vertebra [36]

## 1.3 Shape Analysis Methods:

### 1.3.1 Traditional Methods:

In statistical shape analysis, "multivariate morphometrics" is usually considered as the traditional shape analysis method introduced by C.R. Rao [37]. Later, Reyment et. al. (1984) is given a detailed review of this approach [38]. Multivariate morphometrics is a straightforward approach and performs in two steps. First, this approach suggests selecting a ratio of the distance between landmarks or angles. Second, it requires submitting these ratios to multivariate analysis. Multivariate morphometrics has wide application in shape analysis. Some famous application includes classification of macaque skulls (taxonomy-classification of species) [39] and sexing of Gorilla skulls [8, 40]. In both examples, they used lengths, angles, or ratios of lengths between landmarks, and principal component analysis is used as a common multivariate measurement.

The multivariate morphometrics only considers some positive variables such as lengths, angles, and ratios of the distance between landmarks. This strategy precludes include geometry such as original landmark coordinates. Dealing with actual coordinates is always superior to dealing with former positive variables. Because we can always generate the lengths, angles, and ratios of the distance between angles from actual coordinates where the reverse is not true. In the biological field, multivariate morphometrics is widely used and still a common approach. However. It is always easier to describe an image with original coordinate landmarks that are retaining the geometric approach of an object.

### 1.3.2 Geometrical Methods:

Earlier development of the geometric shape analysis has been done by D'Arcy (1917) Thompson [41] followed by Medawar (1944) [42] and Sneath (1967) [43]. But the major development with lots of varieties had happened since late 1970's and credit goes to D.G. Kendall [32, 44–50] and F.L. Bookstein [51–59]. This vast development allows the researcher to work with actual coordinate landmarks of objects. As statistical shape analysis is inherently non-Euclidean, there is a constraints on landmarks and we are unable to apply standard multivariate techniques on those landmarks.

In that situation, the idea is to work with the complete geometrical objects itself rater dealing with the derived quantities from the organisms [23]. This idea is inspired by D'Arcy Thompson's (1917) geometric transformation of one species to another [Figure 1.6]. Thus Geometrical shape analysis considers a shape space that originated from the

actual coordinate landmarks and retains the geometry of a point configurations [23].

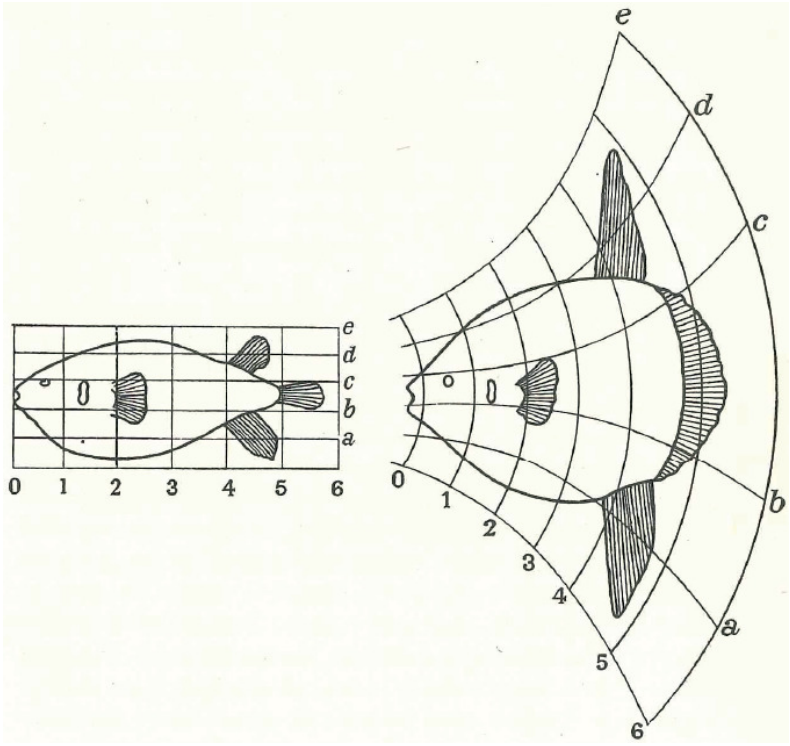


**Figure 1.6.** Geometrically transforming one species of fish, Diodon (the source), into another species, Orthagoriscus mola (the target) [41]

## 1.4 Shape Alignment

In order to get a true shape, we need to filter out location, scale, and rotational effects. This is carried out by setting up a coordinate reference with respect to position, scale, and rotation to which all shapes are aligned. Procrustes analysis [23, 59–61] is an alignment procedure that helps to obtain such coordinate reference. Basically, Procrustes analysis drives the set of shapes into shape space. Different shape alignment procedures are developed so far. We will discuss different methods briefly in subsection 1.4.3.

### 1.4.1 Shape Space

We are adapting the definition of shape space from Dryden and Mardia's statistical shape analysis book [23]:

**Definition 1.4.1.** The *shape space* is the set of all possible shapes of the object. Formally,

the shape space $\sum_k^n$ is the orbit shape of the non-coincident n point set configurations in the $R^k$ under the action of the Euclidean similarity transformations.

But what is the dimension spanned by this shape space? If we have random n point vectors in k Euclidean dimensions the dimensionality is $k \times n$. But the alignment procedure peels of dimensionality, i.e. the data now spans only a subspace of kn. The translation removes k dimensions, the uniform scaling one dimension and the rotation $\frac{1}{2}k(k-1)$ dimensions. Thus, the shape space dimensionality is:

$$M = kn - k - 1 - \frac{1}{2}k(k-1)$$

### 1.4.2 Procrustes Analysis

Procrustes Analysis (PA) is a statistical method used to compare the set of multidimensional shapes or analyze the distribution of a set of shapes [62]. This method attempts to transform the objects into a state of superimposition using optimal isomorphic transformation (translation, scaling, and rotation). Mathematically, PA minimizes the Procrustes distance (sum of squared distances) between corresponding points in each shape of their coordinate matrices [Figure 1.7]. The main goal of the Procrustes Superimposition (PS) is to achieve the similar placement and size of the shapes by minimizing the Procrustes distance between/among the objects.

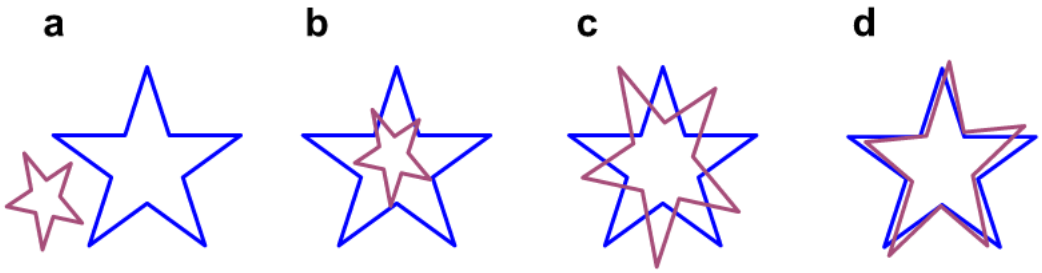


**Figure 1.7.** Optimal Procrustes superimposition of two geometric shapes [62]

It is important to note that PS is of two types: full and partial. Full PS does not consider the scaling transformation as opposed to partial PS. In other words, the object size is preserved in full PS as opposed to partial PS.

Based on the number of shape objects, PA are of two types: ordinary Procrustes analysis and generalized Procrustes analysis. Though our analysis is based on the

generalised Procrustes analysis, we are discussing both for better understating of the process.

#### 1.4.2.1 Ordinary Procrustes Analysis (OPA)

In ordinary Procrustes Analysis, we basically compare two objects each having a finite number of k points in m dimensions. Let us think of each object as a configuration matrix $X$ and $Y$ having dimension $k \times m$ where k represents the number of landmark points having m dimension. It is assumed without loss of generality that both matrices have been centered. According to Dryden and Mardia, ordinary Procrustes analysis completely depends on the Procrustes distance [23].

Procrustes distance is a shape metric that requires two aligned shapes with one-to-one point correspondence. Before computing the Procrustes distance, it is inevitable to align shapes. Shape alignment is a process of superimposing shapes by translating, scaling, and optimally rotating objects. Here, we will discuss the shape alignment procedure considering a simple example:

- **Translation:** Let, both configuration x and y contains k landmark points with dimension $m = 2$: $((x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k))$. The arithmetic mean of these point:

$$\bar{x} = \frac{x_1 + x_2 + \cdots + x_k}{k}$$
$$\bar{y} = \frac{y_1 + y_2 + \cdots + y_k}{k}$$

  Now, we can subtract the mean from each landmark points of the individual objects:

$$(x, y) \rightarrow (x - \bar{x}, y - \bar{y}) = ((x_1 - \bar{x}, y_1 - \bar{y}), (x_2 - \bar{x}, y_2 - \bar{y}), \ldots, (x_k - \bar{x}, y_k - \bar{y}))$$

  Thus, the translational parts is filtered out from the objects.

- **Uniform Scaling:** To measure the objects scale, we can compute the root mean squared distance:

$$s = \sqrt{\frac{\sum_{i=1}^{k}(x_i - \bar{x})^2 + \sum_{i=1}^{k}(y_i - \bar{y})^2}{k}}$$

  Now, we divide the points coordinate by the initial object scale which makes the scale 1.

$$(\frac{x_1 - \bar{x}}{s}, \frac{y_1 - \bar{y}}{s})$$

  This process filtered out the scale components by scaling the objects. We discussed

here RMSD as a shape size metric. One can use other shape size metric such as Frobenius norm or centroid size metric.

- **Rotation:** After removing the translation and scale components, we need to remove the rotation components. Bookstein (19997) applied the Singular Value Decomposition (SVD) technique to get the optimum rotation [59].
  - Step 1: The $k \times m$ dimensional configuration matrix x and y is already aligned with respect to scale and location. We can arrange these matrix. We are considering the most simplaer case of dimension m=2.
  - Step 2: In order to minimize the correlation between two sets of landmarks, we can compute the singular value decomposition (SVD) of $x^T y$. The singular value decomposition is $UDV^T$ where U is a $k \times k$ matrix, D is a $k \times m$ diagonal matrix and V is a $m \times m$ matrix.
  - Step 3: In this step, we can optimally superimpose configuration x upon configuration y by finding the rotation matrix

$$R = VU^T = \begin{bmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{bmatrix}$$

#### 1.4.2.2 Generalized Orthogonal Procrustes Analysis (GPA)

Ordinary Procrustes analysis limits it's scope by comparing only two sets of configurations or shapes. This limitation emerges the scope of developing Generalized Procrustes Analysis which can compare at least two sets of configurations or shapes. The GPA is mandatory while comparing three or more shapes. For matching two shapes we can use either OPA or GPA. In this situation, applying GPA is always advantageous due to it's symmetrical matching characteristics. That is, matching configuration x and y is same as matching configuration y and x. In OPA, symmetrical matching property is only holds if the size of the configuration or shape is same. The GPA algorithm mainly comprises four steps [59, 60]:

- Step 1: Choose a reference shape that is approximate to the mean shape. This can be any shape from the set of available shapes. Researcher often chooses the first shape in the set as a reference shape.

- Step 2: Align all the remaining shapes to the reference shape. The alignment procedure can be done in four steps: First is to calculate the centroid of each shape. In geometric morphometrics, one can compute the centroid (or, center of the gravity) location by averaging the coordinates (say x, y, and z for 3-dimensional shape or configurations) of all the landmark points. After that, each shape's centroid to the origin is aligned following by the normalization of each shape's centroid size. In geometric morphometrics, centroid size refers to the square root of the sum of squared distances of all the landmark points of an object from its centroid. Finally, each shape is rotated to align with the reference shape (approximate mean shape).

- Step 3: From the aligned shapes, recalculate the approximate mean shape.

- Step 4: If the approximate mean shape obtained from step 2 does not match with the approximate mean shape obtained from step 3, return to step 2 and repeat the process. Otherwise, the process will provide a true mean shape of the set.

Usually, calculation of the true mean shape is computationally challenging in terms of program execution time and computer memory. Computational complexity increases many times if the number of objects in a data set is high. The mathematical background of the shape transformation (translation, isomorphic scaling, and rotation) for more than two shapes is discussed below:

- **Translation:** Translation step translates all the shapes to a common center. Mathematically, we can translate shapes as follows:
  Let, Z be a $k \times m$ matrix where k is the landmark points having m dimensions. Now, we can compute the centroid matrix by taking the arithmetic average of the columns of the Z matrix. Subtracting the centroid matrix $Z_c$ from Z will give the new coordinate of Z centered at the origin, denoted by $Z_o = Z - Z_c$.

$$Z_o = Z - Z_c = \begin{bmatrix} z_{11} - \bar{z_{i1}} & z_{12} - \bar{z_{i2}} & \dots & z_{1m} - \bar{z_{im}} \\ z_{21} - \bar{z_{i1}} & z_{22} - \bar{z_{i2}} & \dots & z_{2m} - \bar{z_{im}} \\ \vdots & \vdots & \dots & \vdots \\ z_{k1} - \bar{z_{i1}} & z_{k2} - \bar{z_{i2}} & \dots & z_{km} - \bar{z_{im}} \end{bmatrix}$$

- **Isomorphic Scaling:** Isomorphic scaling is a transformation technique used to scale the shapes to a similar size. More specifically, to maintain the ratio of the

shape, this technique makes the shape of an object small or big. One of the popular isomorphic scaling techniques is Normalization. Mathematically, the centered and normalized coordinate matrix $Z_n$ can be defined as,

$$Z_n = Z_o\Big(\frac{1}{||Z_o||}\Big)$$

- **Rotation:** After translating and scaling the shapes, rotation transformation basically aligning all the shapes to a target shape. Target shape can be an average of all the shapes or any shape from the data set. In this discussion, we are considering average shape as a target shape. The rotation technique uses the Euclidean or Frobenius method to minimize the sum of squared distances between the target shape matrix and rotated shape matrix as follows [63]:

$$minmize||Z_nQ - \bar{Z}|| \tag{1.1}$$

implies,

$$minimize[trace((Z_nQ - \bar{Z})'(Z_nQ - \bar{Z}))] \tag{1.2}$$

implies

$$minimize[trace(Z_n'Z_n + \bar{Z}'\bar{Z}) - 2 * trace(\bar{Z}'Z_nQ)] \tag{1.3}$$

Now to minimize (1.3), we need to maximize $trace(\bar{Z}'Z_nQ)$. Applying singular value decomposition of $\bar{Z}'Z_n = USV'$, we can write

$$trace(\bar{Z}'Z_nQ) = trace(USV'Q) = trace(SV'QU) = trace(SH) \tag{1.4}$$

where H is the $p \times p$ orthogonal matrix. Thus,

$$trace(SH) = \sum_{i=1}^{p} s_i h_{ii} \tag{1.5}$$

Since, $s_i \geq 0$, trace (SH) is maximized when $h_{ii} = 1$ for $i = 1, 2, \ldots, p$. Thus,

$$H = I = V'QU$$

Thus, $Q = VU'$ minimizes (1.1). Therefore, we can tell that the rotation process is completed by multiplying $VU'$ to the $Z_n$ matrix in order to align with the $\bar{Z}$.

This theoretical background of Procrustes analysis is portrayed in Figure 1.8.

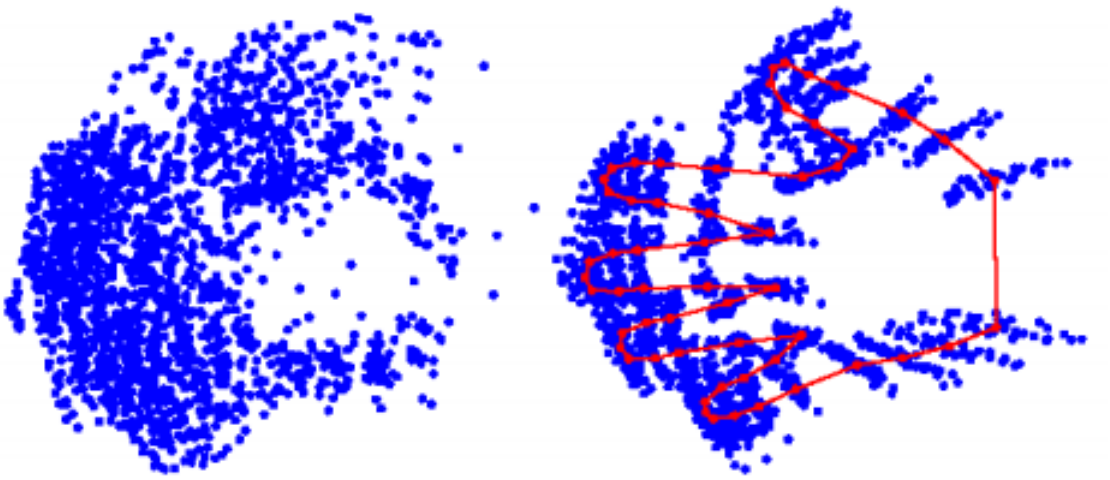

**Figure 1.8.** Left: 40 unaligned shapes. Right: 40 aligned shapes with the mean given in red [64]

### 1.4.3 Different Shape Alignment Methods

There exits a number of shape alignment methods widely used in the filed of compute vision and image processing. In this subsection, we will only mention few of them. Davies [86] has used an information-theoretic framework to formulate shape alignment as a minimum description length (MDL) problem. The main advantage of this method is that it can register multiple shapes simultaneously without the requirement of the template shape or target shape. It is also note that this advantage turned to be a limitation when atlas is present for a particular problem where all data shapes are to be registered to it. Granger [87] developed the EM-ICP method, types of the famous ICP algorithm mainly used in the rigid shape alignments. The main limitation of the ICP method is that it terminates the algorithm at suboptimal local minima, which is greatly reduced by treating the problem as a general expectation-minimization problem in EM-ICP. Few previous methods used spherical harmonics for rigid shape alignments such as first order ellipsoid (FOE) introduced by Brechbuhler [88]. This method is widely used in the medical imaging application [89, 90]. This alignment procedure only use first order spherical harmonics to reconstruct shapes, which forms an ellipsoid in object space. This method works its best when the length of the three axis of the ellipsoid is different. This method provides crude alignment and fail if two or more axis has similar lengths. SHREC [91] is another type of ICP where Rigid Quaternion Transform (RQT) is used to align shapes in R3 after each iteration of parametric alignment. As the optimization

in parameter space depends on the objects position, this method is not guaranteed to converge to the optimal solutions. Another important approach for shape matching and comparison is optimal mass transport. This surface based 3D shape analysis approach plays a key role in computer vision and medical imaging [92].

## 1.5 Theoretical Background: Deterministic Atlas

The template shape corresponds to a mean of the given objects. In the atlas model, we are estimating the mean object from a group of objects and the deformation from this mean object to each object in the group [65–68]. This process is known as atlas construction. The mathematical background of this atlas construction is based on the Large Deformation Diffeomorphic Metric Mapping (LDDMM) framework. We will discuss the LDDMM algorithm in the light of Deformetrica.

### 1.5.1 Large Deformation Diffeomorphic Metric Mapping (LDDMM)

The LDDMM algorithm uses diffeomorphic transformations of the 2D or 3D ambient space between them to compare the shapes. We must do the following two steps to perform LDDMM: To parametrize a large family of transformations of the 2D or 3D space and compute distances between the shapes (meshes or images).

#### 1.5.1.1 Parametric Family of Deformation: Constructing Diffeomorphisms

Deformetrica provides a complete parameterization of diffeomorphisms of the low dimensional ambient space $R^d$ where $d = 2, 3$. Let, $(c_i)_{i=1,2,3,\ldots,p}$ be the set of p "Control" points and $(m_i)_{i=1,2,3,\ldots,p}$ be the set of p "Momentum" vectors of $R^2$ or $R^3$. Deformetrica uses this set of p control points and momenta to form a vector field on the 2D or 3D space. The vector field can be denoted as V and defined as,

$$V(v) = \sum_{i=1}^{p} K(v, c_i).m_i \tag{1.6}$$

Here, K is a Gaussian kernel with width $\sigma$ which can be defined as, $K(v, c_i) = exp(-\frac{||v-c_i||^2}{\sigma^2})$. The vector field is assessed at point v. Here, parameter $\sigma$ controls the deformation and is called the critical hyperparameter in any deformation. That is the large value of $\sigma$ can generate smooth and global deformations. In contrast, a small value of $\sigma$ generates deformations with a thin level of precision. In Deformetrica, $\sigma$ is

introduced in the `model.xml` script file as `deformation parameters` will be introduced in the section 1.3. A general thumbs-up rule for defining the value of $\sigma$ is that irrespective of the situation, the starting value of $\sigma$ should be low and gradually increase if needed.

Equation (1.7) represents the Reproducible Kernel Hilbert space (RKHS) H with norm:

$$||V||_H^2 = \sum_{i,j=1,2,...,p} K(c_i, c_j).m_i^T m_j \tag{1.7}$$

A time dependent equation for both control points and momentum sets are suggested which are also known as Hamiltonian equation:

$$\begin{aligned} \dot{c}(t) &= K(c(t), c(t)).m(t) \\ \dot{m}(t) &= \frac{-1}{2}\nabla_c\{K(c(t), c(t)).m(t)^T m(t)\} \end{aligned} \tag{1.8}$$

Either Euler or Runge-Kutt scheme of order two is used to integrate the above-mentioned Hamiltonian equation. This way, a vector field $V(v, t)$ can be generated, which varies over time. This time-varying vector field is calculated using equation (1.6) with respective control points $c(t)$ and momenta points $m(t)$.

Let us consider $v \in R^d$ be a point of the ambient space. Now, at time $t = 1$ we can define the transformed point $\Phi(v)$ of the function $L : [0, 1] \mapsto R^d$ with starting condition $L(0) = v$. This function carry out the ordinary differential equation:

$$L'(t) = V(L(t), t) \tag{1.9}$$

The obtained mapping $\Phi : R^d \mapsto R^d$ which is a diffeomorphism of the ambient space $R^d$. A detailed mathematical discussion of this procedure is available in L. Younes's Shapes and diffeomorphisms article [30].

Overall, the obtained diffeomorphism $\Phi$ is fully parameterized by initial sets of control points c and momenta m: we will note $\Phi = \Phi_{c,m}$. This simple parametrization of a large family of diffeomorphisms paves the way to optimizing the initial control points c and momenta m to estimate a desired transformation of the ambient space. Consequently, each model will be formulated as an optimization problem on these variables.

The deformation of a mesh or an image under such a deformation is computed by a convolution between the control points and their momenta and the meshes' vertices or the voxels of the images. It has a complexity in $n \times d$ where n is the number of control points, and d is the number of vertices/voxels of the shape. Consequently, on large objects, the use of GPU computations provides a sensible speed-up.

#### 1.5.1.2 Distance between Shapes: Meshes or Images

A metric is needed to evaluate if the deformed shape is close to its target. We will discuss the distances of shapes for mesh data and image data separately.

(a) Distance between images: In Deformetrica, the Euclidian $l^2$ distance is available for measuring the distance between two images. This distance is the sum of the square of the voxel-wise differences between two images.

(b) Distance between meshes: A mesh is nothing but a collection of polygons. Any objects created by polygon meshes must include various parts such as vertices, edges, faces, polygons, and surfaces. The faces of different objects are in different geometric shapes such as faces of `SurfaceMesh` object is a triangle whereas faces of `PolyLine` objects are segments. In both cases, we can calculate the centers and normals of the edges. Let us define,

$C_s$ = Center of the edges where $s = 1, 2, \ldots, g$

$N_s$ = Normals of the edges where $s = 1, 2, \ldots, g$

$V_s$ = Vertex of the mesh where $s = 1, 2, \ldots, d$

For meshes, we can use the Euclidean $l^2$ distance metric if there is point-to-point correspondence. Without point correspondence (in general case), we can use the "current" or "varifold" distances.

- Current Distance: Equipped with centers and normals of the edges, we can compute the current distance between two meshes as follows:

$$d((C_s^\gamma, N_s^\gamma)_{s=1,2,\ldots,g^\gamma}, (C_t^\delta, N_t^\delta)_{t=1,2,\ldots,g^\delta})^2 = \sum_s \sum_t G_k^Z(C_s^\gamma, C_t^\delta)(N_s^\gamma)^T N_t^\delta$$

  where $G_k^Z$ is the Gaussian kernel having width $\sigma_k^Z$

- Varifold Distance: Also, we can compute the varifold distance which ignores the orientation of the normals:

$$d((C_s^\gamma, N_s^\gamma)_{s=1,2,\ldots,g^\gamma}, (C_t^\delta, N_t^\delta)_{t=1,2,\ldots,g^\delta})^2 = \sum_s \sum_t G_k^Z(C_s^\gamma, C_t^\delta)\frac{((N_s^\gamma)^T N_t^\delta)}{||N_s^\gamma||.||N_t^\delta||}$$

In Deformetrica, we set the hyper-parameter $\sigma_k^Z$ in the `model.xml` file for each mesh object. It is desired to have a small value of $\sigma_k^Z$. A small value of $\sigma_k^Z$ indicates that the fit is very close. On the other hand, a large value of $\sigma_k^Z$ indicates that the fit is globally good. A good practice to start with a large value of $\sigma_k^Z$ (usually one-

fifth of the length of the mesh) [30,31]. Then gradually trying the smaller values if needed.

Sometimes, the template object is composed of several different objects. In such a situation, Deformetrica allows computing the concurrent deformation of multiple shapes which can be embedded in the same ambient space $R^d$ [31]. Thus Deformetrica can compute the square distance between two objects $O^\gamma$ and $O^\delta$ as follows:

$$d(O^\gamma, O^\delta)^2 = \sum_{r=1}^{n_k} \frac{d(S_r^\gamma, S_r^\delta)^2}{\sigma_r^2} \tag{1.10}$$

Where, $O^\gamma = (S_1^\gamma, \ldots, S_{n_k}^\gamma)$ and $O^\delta = (S_1^\delta, \ldots, S_{n_k}^\delta)$ are two objects. Each object consists of $n_k$ homologous shapes.
The distance metric is the weighted average of the squared distances of the corresponding objects. The parameters $\sigma_r$ can be used to tune the relative importance of each part of the composite "multi-object" of study.

#### 1.5.1.3 Loss function for the Deterministic Atlas Model

This model can be formulated as an optimization problem on a loss function. Let us consider a collection of shapes $S_i; i = 1, 2, \ldots, n$. The Atlas model calculates the mean (M) of these shape and a set of momenta (diffeomorphisms) $\Phi_i; i = 1, 2, \ldots, n$ such that each shape is well reconstructed as follows:

$$M * \Phi_i \simeq S_i; i = 1, 2, \ldots, n$$

This is achieved by the minimization of the loss function:

$$f(M, q, \mu_i) = \sum_{i=1}^{n} d(\Phi_{q,\mu_i} * M, S_i)^2/\sigma_\epsilon^2 + R(q, (\mu_i)) \tag{1.11}$$

with

$$R(q, \mu_i) = \sum_{i=1}^{n} \mu_i^T K(q, q)\mu_i$$

where, K(q,q) is the $p \times p$ kernel matrix $[K(q_s, q_t)]_{s,t=1,2,\ldots,p}$.

The first term in equation (1.11) controls the data attachment. More specifically, it measures how well the set of objects is estimated by the deformation of the template where the second term acts as a regularizer by inflicting a penalty on the kinetic energy of the deformations [31]. The relative importance of those two terms is specified by the

user through the parameter $\sigma_\epsilon$.

## 1.6 Theoretical Background: Geodesic Regression

Geodesic regression basically generalizes linear regression to a context with a manifold-valued response variable and one or more real-valued independent variables. For example, the manifold valued dependent variable would be 2D or 3D landmark-based facial objects or images where the real-valued independent variable would be age, height, or weight of a facial object or time recorded from repeated MRIs of the same individual. The objective of the geodesic regression use the set of observed shapes $(S_i; i = 1, 2, \ldots, n)$ at time $t_i$ within the time interval $[t_0, T]$ in an intention to compute the continuous evaluation of shape which best describes the observed data (Figure 1.9).

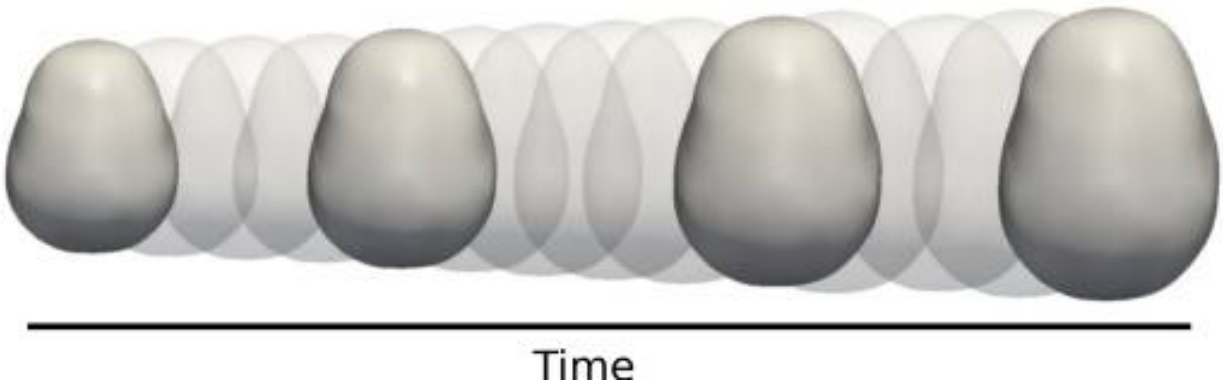


**Figure 1.9.** Shape regression used four time indexed observations (solid surfaces) to estimate the continuous shape evaluation (transparent surfaces) [69]

Geodesic regression requires two important components such as deformation models and parameterization of the time-varying deformation $\phi_t$. We discussed the large deformation model LDDMM and the family of parameterization in the previous section (section 1.5). Here, we will discuss the geodesic model which can be formulated as an optimization of the loss function:

Let us consider a collection of shapes $(S_i; i = 1, 2, \ldots, n)$ at time $t_i$ within the time interval $[t_0, T]$. The shape evaluation can be modeled as a geodesic flow of diffeomorphisms $(\phi_t)$ applied on template shape or mean shape (M) such that each shape is well reconstructed as follows [69]:

$$M * \Phi_t \simeq S_i; i = 1, 2, \ldots, n$$

where t varies continuously within the time interval determined by the observed data. That is, over time the template shape M is continuously deformed in an intention to match the observed shape data $S_i$.

The cost function for Geodesic regression:

$$f(M, q, \mu) = \sum_{i=1}^{n} d(\Phi_{q,t_i.\mu} * M, S_i)^2/\sigma_\epsilon^2 + R(q, \mu) \tag{1.12}$$

with

$$R(q, \mu) = \sum_{i=1}^{n} \mu_i^T K(q, q)\mu_i$$

where, K(q,q) is the $p \times p$ kernel matrix $[K(q_s, q_t)]_{s,t=1,2,\ldots,p}$.

The data attachment is controlled by the first part of equation 1.12 where the second part penalizes the kinetic energy of the deformation [31]. Here, the user-specified parameter $\sigma_epsilon$ is a hyperparameter that basically addressed the tradeoff between data attachment and regularity. Minimization of the cost function produces a template shape M, a set of control points, and associate initial momenta so that the geodesic trajectory $t \mapsto \Phi_{q,t_i.\mu} * M$ is as close to the input shape data. Alexandre Bône et. al. discussed an example of the estimated geodesic regression [Figure 1.10] which is based on 3D meshes of human faces [31].

**Figure 1.10.** Estimated geodesic regression. Top row: the estimated trajectory. Bottom row: observations from which the top trajectory is learned. [31]

## 1.7 Tools and Techniques: Atlas Construction

Deformetrica is a shape analysis software developed by the Aramis Lab team in Paris. Deformetrica supports different types of objects such as Landmark, Polyline, SurfaceMesh,

Point clouds, Image, or any combination of them. It computes deformations of the 2D or 3D ambient space, which, in turn, enfolds any of the above objects embedded in this space. Deformetrica is an excellent choice to perform atlas construction as it has low data requirements.

### 1.7.1 Deformetrica Installation:

Deformetrica installation can be done in several steps:

- Step I: The basic requirements for installing Deformetrica are Anaconda 3.0, Linux, or Mac OSX distributions for Python 3.6, 3.7, and 3.8. Our study installed Anaconda 3 and used Deformetrica 4.3.0 Rc available on Linux for Python 3.8.
- Step II: Install Conda or Miniconda.
- Step III: Installation of Deformetrica is available through two different channels: a) Repository and b) Conda setup. We used conda setup method as it is relatively simple. Following conda command will create and activate conda environment and install Deformetrica as well as its dependencies:

  ```
  $ conda create -n deformetrica python=3.8
  $ source activate deformetrica
  $ conda install -c pytorch -c conda-forge -c anaconda -c
  aramislab/label/rc deformetrica
  ```

- Step IV: Check the status of the installation using the command:
  `$ deformetrica -help`

Due to the large volume of data and high computational burden, we performed installation steps in Roar Supercomputer at Penn State using GPU based allocation queue. From a personal computer, the Deformetrica installation is easy and straightforward. For an extensive data set, a supercomputer is a mandatory option. Installing Deformetrica is a bit complicated while using a high-performance computing server as lots of files and dependencies installation requires permission.

### 1.7.2 Configuration File Set-up:

Before we run the atlas estimation, we need to set up three configuration xml files described below:

1. `model.xml` file: The `model.xml` file must contain the following parts:

    - Model-type: DeterministicAtlas
    - Dimension: 2 for 2D images and meshes, 3 for 3D images and meshes
    - Template: Contains following parts which helps to describe the objects used in computation:
        - deformable-object-type: Any one of the following object type: Landmark, Polyline, SurfaceMesh, Point clouds, Image, or any combination
        - attachment-type: In Deformetrica, attachement-type means type of distance used between objects. Distance type is different for different objects. For instance, if the object is image, landmark and pointcloud we don't require attachment type as it measure ecludian distances ($l^2$ distances). For Polyline, the attachment type is Verifold.
        - kernel-type: Two types of kernel are used: kernel operations (keops) and torch. Kernels are mainly used to perform deformations. If the objects are large, it is recommended to use keops as it solves the memory overflow problem.
        - noise-std: It measures how well we want this object to be fit. Low value of noise-std provide more importance to the data attachment cost for this objects.
        - filename: Indicates a file that can be used as an initialization. It can be a template/ mean shape or any observation from a set of objects.
    - Deformation-parameters: Mainly specifies the type of deformation used in the algorithm.
        - kernel-width: This is the only mandatory option as a deformation parameter. It is used in the convolutions to generate deformation. It is desirable to start with a large value of kernel.

2. `data_set.xml` file: The `data_set.xml` is formed in an organized way. The file contains mainly three parts: `<subject>, <visit>, <filename>`. `<subject>` are the first level of hierarchy. One subject can visit one or several times. `<visit>` option indicates each visit separately. Each visit contains one or several structures identified through `<filename>` option. The atlas construction requires a cross sectional data which allows a single visit for each subject.

3. `optimization_parameter.xml` file: `optimization_parameter.xml` has lots of options. We are discussing here some key options that is enough to perform the estimation:

    - optimization-method-type: Different optimization methods are `GradientAscent`, `ScipyLBFGS` or `McmcSaem`. The default one is `GradientAscent`. The Deterministic Atlas approach uses only `GradientAscent` and `ScipyLBFGSor`.
    - max-iterations: The default value is 100.
    - max-line-search-iterations: One can specify the max-line-search-iterations for `GradientAscent` estimator.
    - convergence-tolerance:
    - gpu-mode: can take the value 'auto', 'none', 'full' or 'kernel'. 'full' will run all computations on a gpu whereas 'kernel' will only run the kernel convolutions on gpu.
    - initial-step-size: The default initial-step-size value is $1e^{-3}$. This value need to make lower if the line-search iterations failed.
    - number-of-processes: This option allows single processes and multi processes. Not all application supports multi process (e.g., Bayesian Atlas allows only single process). Multi process is only supported by Deterministic Atlas and Longitudinal Atlas.

### 1.7.3 Atlas Estimation:

After setting up all the configuration files, we need to input the xml files and optionally setting the verbose setting to INFO. Finally, we can run the Deformetrica estimate command to perform the actual atlas estimation:

```
$ deformetrica estimate model.xml data_set.xml -p
optimization_parameters.xml -v INFO
```

### 1.7.4 Output files

An output folder will be created in the current working directory where the folder will contain different types of output files:

- `log` file: Date and time stamped console output information such as all informations related to iteration steps (likelihood values, step size and gradient norms etc.) and program execution time.
- `.vtk` file: Few .vtk files such as final deformation file or reconstructed files, intermediate deformation step files and estimated template file which corresponds to the mean template file
- `.txt` files: Few .txt files such as estimated momenta, estimated control points and estimated residuals
- `deformetrica-state.p` file: This file can be used to resume the estimation

### 1.7.5 Parallel Computing: CUDA Acceleration

Compute Unified Device Architecture (CUDA) is a parallel computing platform invented by Nvidia (`https://developer.nvidia.com/cuda-zone`). It allows using a CUDA-enabled graphics processing unit (GPU). Deformetrica can automatically detect CUDA device. If CUDA is available in the system, the kernel operations will be conducted on the GPU. In our study, we used NVIDIA TESLA K80 GPU for getting sensible speed in the mean face computation.

## 1.8 Tools and Techniques: Estimating Shape Trajectory

### 1.8.1 Configuration file set-up

In subsection 1.7.1, we described the installation procedure of Deformetrica. After installation, we need to set up different configuration xml files as follows:

1. `model.xml` file: This file has same architecture as Deterministic Atlas except the following four options which mainly determines the program execution success:
    - Model-type: model has to be “Regression".
    - concentration-of-time-points: Mainly used for controlling the discretization steps of a geodesic. By-default deformetrica takes 10 as its value. But, one can take lower or upper value based on the problem. A higher value will give more descritization steps where a smaller value will provide less descritization steps. In our problem, concentration of time point was 1.

- t0: This option controls the point from which the geodesic is computed. This value should initialize with great care because the given initial template will be the point on the geodesic at t0. If we don't define this point, Deformetrica will consider the mean age as its default value.
- template-frozen: During geodesic regression this option is on that is template is frozen and it is advised to keep template frozen as the estimation of the momenta of the geodesic regression and the template can be ill-defined.

2. `data_set.xml` file: The `data_set.xml` file contains 4 parts: `<subject>, <visit>, <age>, <filename>`. The only addition with the previous one is `age` which specifies the corresponding time of each observation/geometric objects.

3. `optimization_parameter.xml` file: This file has same structure as Deterministic Atlas. One can fine tune the options based on the problem.

### 1.8.2 Geodesic regression estimation

Once set-up all configuration files, one can run the deformetrica estimate to perform the geodesic regression estimation:

```
$ deformetrica estimate model.xml data_set.xml -p
optimization_parameters.xml -v INFO
```

### 1.8.3 Output files

Like Deterministic Atlas, an output folder will be created in geodesic regression estimation having different output files:

- `log` file: Contain same type of information as Determinstic Atlas.
- `.vtk` file: Few .vtk files such as final deformation file or reconstructed files, intermediate deformation step files which is geodesic flow files correspond to different concentration of time points and estimated template file correspond to average time point used for initialization.
- `.txt` files: Few .txt files such as estimated momenta and estimated control points
- `deformetrica-state.p` file: Same functionality as Deterministic Atlas.

## 1.9 Benefits of the Study:

Shape averaging or computing means shape from a set of shapes builds a generalization of the typical representation. It is assumed that the mean shape can preserve the original shapes' attributes and thus useful in forecasting trends in form or pulling out stereotypes from a set of homologous shapes. Our study serves three purposes to compute the mean shape: First, it provides a comprehensive description of the Deformetrica software regarding applying the Deterministic Atlas model and related data processing tools and techniques. Secondly, it discusses the implementation of the Deterministic Atlas model on 3D Facial Norm data-set using GPU-enabled computing. Lastly, it helps to perform the Geodesic Regression considering mean shape (obtain from Deterministic Atlas model) as a template shape (in Geodesic Regression Model) where the geodesic regression modeled the continuous shape evaluation over time.

## 1.10 Contribution of the Study:

The prime contribution of this study is that for the first time it implements a sophisticated shape analysis software Deformetrica on 3 dimensional facial norm (3DFN) data [70] deposited in the FaceBase Consortium [72]. A number of studies have been conducted based on different shape data deposited in the Facebase project [71, 93–97] but no one explores those data (including 3DFN) using Deformetrica. A detailed description of the 3DFN dataset is provided in chapter two. Previous studies show that Deformetrica was applied on different shape models having low number of landmark points [69, 98]. This study implemented and discussed the scope of dealing shape models having very high number of landmark points (accessing GPU).

# Chapter 2
# Results

## 2.1 Data

This study uses the 3D Facial Norms database (3DFN), which provides craniofacial anthropometric normative data [70]. 3DFN database consists of 2454 male and female participants satisfying two inclusion criteria: I) European Caucasian ancestry and II) Age between 3 to 40 years. The FaceBase project recruited the participants in three sites in the United States (the University of Pittsburgh, Seattle Children's Hospital, and University of Texas Health Science Center at Houston). The database is available at both summary-level and individual-level. Summary-level data provides summary statistics of different demographic variables (e.g., age, height, etcetera) and demographic variable-specific anthropometric measurements (e.g., an intercanthal distance of 10-year-old females). On the other hand, individual-level data includes summary-level data plus landmark coordinates and individual facial surface files [71]. All individual-level data are currently controlled-access. We completed all the necessary steps for accessing controlled human data on FaceBase projects [72]. Our study used 550 3D facial surface files from the 3DFN database provided in object wavefront (.obj) format. All surface files are without the associated color/texture mapping, meaning that one can see the geometry without color and texture associated with subjects' faces.

## 2.2 Data Processing

The original data is in object wavefront (.obj) format, where Deformetrica only allows the visualization toolkit (.vtk) format. So, we wrote an R program to convert the obj format files into vtk format. There are different 3D visualization software, including free

programs like MeshLab, which can clean the meshes [73, 74]. We used MeshLab to clean the mean face by doing hole-filling, mesh-repairs, and smoothing routines. As mean face can be used as a template face in the geodesic regression program using Deformetrica, cleaned mean face will help get a better estimation result. One of the crucial things is that MeshLab does not open any vtk file. Thus, after getting the mean face in vtk format generated by Deformetrica, we again converted the vtk face in obj format. We wrote a MATLAB code that converted the vtk into obj format. It is important to note that, for viewing the 3D surface mesh vtk files, we used ParaView, an open-source, multi-platform data visualization application [75, 76]. ParaView also helped us set the 3D face and change the model color and the background.

## 2.3 Application of Deformetrica: 3D Facial Norms Atlas Construction

We divided our data set into 11 equal groups or clusters, and each has 50 3D facial norm objects. We run the Deterministic Atlas model over each group, which provides a mean face/atlas. Further, we run our Deterministic Atlas model over 11 mean faces and finally got the desired mean face. We are showing here the cleaned mean face for each cluster [Figure 2.1] and the overall mean face [Figure 2.2].

An important issue while performing the Deterministic Atlas model is memory and execution time. The `keops` kernel controls the memory overflow problem but program execution time was still a problem. We solved this problem by using the GPU allocation. We run the program on 50 faces without the GPU allocation and the program was terminated due to time restriction (48 hours). Though implementing GPU allocation provides a sensible speed, it was still challenging to run the program on all the faces. For easier computational aspects, we divided the data into 11 equal clusters and run the program. Table 2.1 shows the execution time for each cluster and overall data.

After getting this overall estimated mean face, our objective was to compare this to the Euclidean mean face (only averaging the coordinate points for actual data) to measure how close they are. As our 3D surface models contain different numbers of landmark points, we could not compute the Euclidean mean face, and subsequently, the comparison was not possible. Another important application of estimated mean face is that one can use it as an initial template face to initiate the Geodesic Regression Program (another important application of Deformetrica).

## 2.4 Application of Deformetrica: Estimating Geodesic Regression

We used five 3D facial norm objects from 3DFN database [70] to perform the geodesic regression. While running the geodesic regression we had no longer access to the GPU allocation. As a result, Deformetrica took a huge amount of time even for a small number (only 5 faces) of facial objects. Thus, we decided to present our geodesic regression results based on a small number of facial norm objects.

Before running the geodesic regression, we performed the Deterministic Atlas model to get the mean face based on these five objects. We used this mean face as the initial template face or baseline shape. This program takes five faces ($S_i$) with corresponding ages ($t_i$) as input data.

Once the program complete, the output folder contains the estimated momenta and control points, as well as the whole geodesic trajectory and the reconstructed targets. Figure 2.3 shows the estimated geodesic regression. Figure 2.3 shows a shape evaluation over time learned from an observed facial object. It is also evident that the geodesic shape trajectory is as close as to the observed data. The estimation took 20 hours, 47 minutes, and 14 seconds. It is also noted that the geodesic trajectories deformation level is higher than the expectation. More appropriate values chosen for the deformation parameter will give a closer shape trajectory to the object.

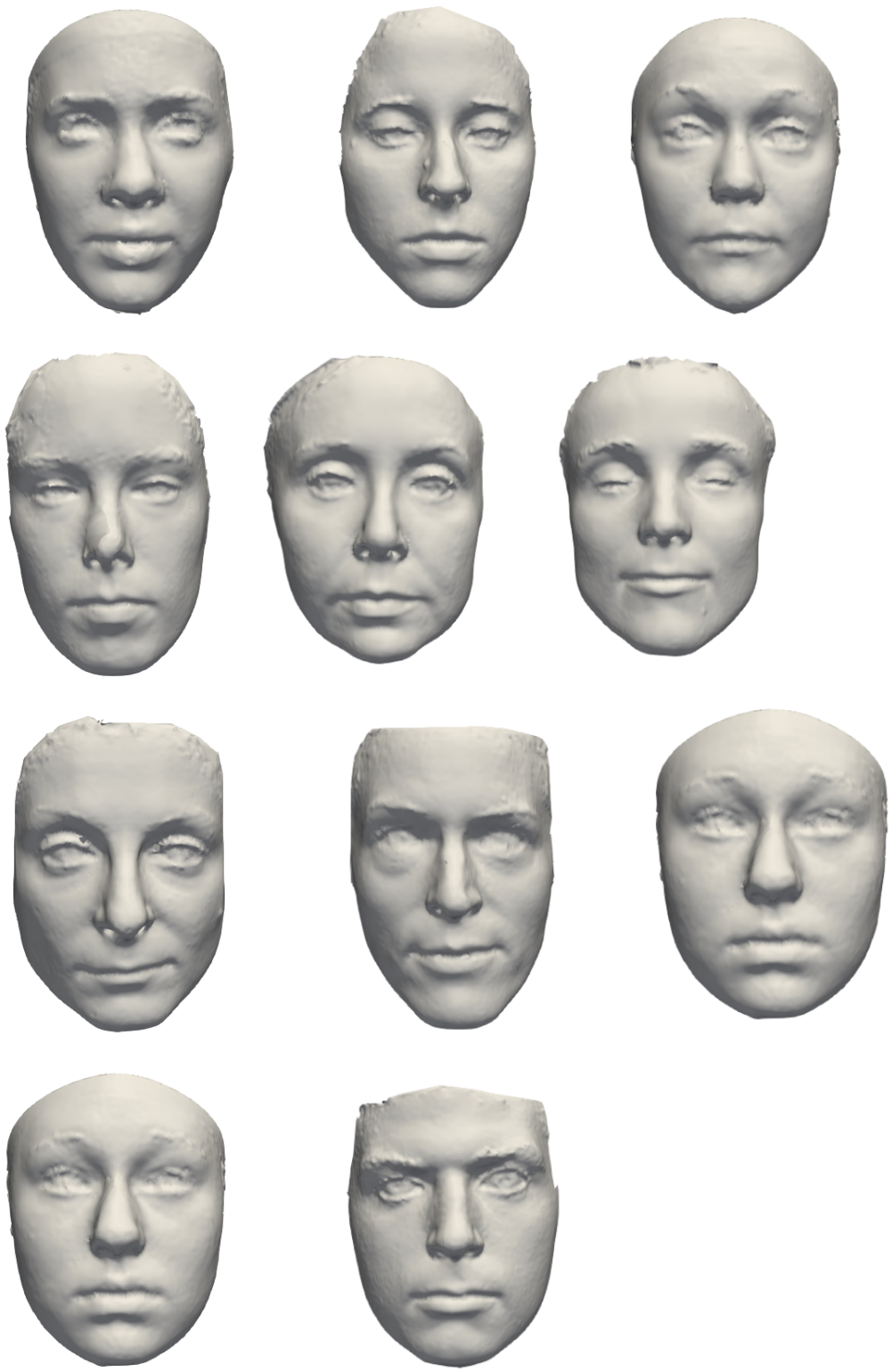

**Figure 2.1.** Eleven cleaned 3D Mean Face based on 11 clusters each having 50 objects

**Figure 2.2.** A cleaned 3D Mean Face based on 550 facial norm shape objects (or, based on 11 cluster mean face objects)

**Table 2.1.** Program execution time using GPU allocation

| Clusters | Execution time |
|---|---|
| 1 | 05 minutes and 16 seconds |
| 2 | 08 minutes and 53 seconds |
| 3 | 04 minutes and 36 seconds |
| 4 | 15 minutes and 52 seconds |
| 5 | 06 minutes and 09 seconds |
| 6 | 05 minutes and 14 seconds |
| 7 | 07 minutes and 40 seconds |
| 8 | 06 minutes and 39 seconds |
| 9 | 05 minutes and 17 seconds |
| 10 | 15 minutes and 21 seconds |
| 11 | 07 minutes and 54 seconds |
| overall | 3 minutes and 34 seconds |

**Table 2.2.** Faces with corresponding ages

| Face | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| Age | 18 | 22 | 33 | 35 | 11 |

**Figure 2.3.** Estimated geodesic regression. First two rows: Estimated geodesic trajectory or continuous evaluation of 3D facial norm shape. Last row: Observed data from which the top trajectory is learned

# Chapter 3
# Conclusion

## 3.1 Discussion

This study presented the fast implementation of the Deterministic Atlas model of large data sets in the LDDMM setting using the Deformetrica. It also provides an in-depth overview of the Deterministic Atlas model's computational aspects using GPU-based parallel computing. We investigated this 3DFN data set in various ways. When we started with a small data set (5 3D faces), the computation was done by Deformetrica using `torch` kernel and does not require GPU access. When we started increasing the number of data points (Say, 20 3D faces), we first encountered the memory problem as `torch` kernel is too memory greedy. In that situation, we changed the kernel to the `keops`. That sharply solved the memory overflow problem. Thus it is highly recommended to use the `keops` kernel for convolutions with several thousands of points. Secondly, we encountered a time problem while started using `keops` kernel. For instance, running the program on five 3D faces took 4 hours 30 minutes, where running the same program on 20 faces exceeded the default time limit (48 hours) in an open queue. To solve this problem, we run the GPU compute allocation program from the Roar Supercomputers at Penn State. The GPU allocation provides us with a sensible speed. For Instance, constructing a mean face that takes 3 to 8 minutes based on 50 3D faces for 11 clusters or groups.

In this study, we used NVIDIA Tesla k80 GPU as it was the only available item while purchasing. There are more powerful GPUs available. Using NVIDIA Tesla P100 or V100 will provide more speed executing the Deterministic Atlas model. Most of the Medical and anthropocentric data are on a large volume scale. As the Deterministic Atlas model computation is highly intensive because of the nature of the data, it is highly recommended to accelerate the computation from CUDA capable GPU. Though

Deformetrica has very low data requirements, it requires purely geometrical modeling of the shapes. That is a deformation model cannot change the topology of the deformed image [9].

Before running Deterministic Atlas program on our 3DFN data set, we run this program on different image and landmark based data set available in the GitLab (`https://gitlab.com/icm-institute/aramislab/deformetrica.git`). The landmark based 2D Skull data set contains between 150 and 200 landmark points while 3D Surprise datset contains between 600 and 1000 landmark points. Both data set contains a low number of landmark points which makes the estimation less memory consuming and less time consuming. In both cases, we used `torch` kernel and kernel device cpu instead of gpu. Both program complete the Gradient Ascent optimization within 1 minutes without showing any memory issues. In that situation, our experience tells that if we have large number of objects (image or mesh) with low number of points, Deformetrica can compute the atlas within short time without the memory issues having kernel `keops` and GPU allocation. But A high number of points having small number objects (in our case 3DFN data) will face memory overflown problem with high amount of computational time. In that case, memory problem still can be solved using `keops` kernel but the computation time still a major challenge to solve.

To perform geodesic regression using Deformetrica, we first considered all 550 facial norm objects. But, due to high landmark points (between 19000 and 50000) and unable of accessing the GPU allocation, we decided to run the program on each individual cluster having 50 faces. But, the program was killed because of the slow speed. We solved the memory overflown problem using `keops` instead of `torch` but we could not solve the time problem. The GPU allocation can provide a sensible speed but still, it can't handle the situation well when the object has a very high number of landmark points. Like the Deterministic Atlas model, we run the geodesic program using Deformetrica on different 2D and 3d datasets available in the Gitlab (`https://gitlab.com/icm-institute/aramislab/deformetrica.git`). All those example data sets have very few landmark points that help to complete the Gradient Ascent estimation within one minute where the program used the `torch` kernel without having any GPU allocation. But, due to the high time-consuming nature of the data, we just choose a small subset of faces from the overall sample. We presented the geodesic flow of diffeomorphisms which basically used to model the shape evaluation acting on a template shape obtained from the deterministic atlas computation [69].

## 3.2 Limitation

We need to utilize tools to automate the cleaning. If the data set is large enough, it is highly tedious and sometimes impossible to manually clean all the geometric shapes. Running a Deterministic Atlas or Geodesic regression program based on all clean geometric shape data points will provide a better estimation. Another drawback of this study is that we found the mean face based on the cluster/groups because of the memory problems. Despite the use of `keops` kernel, we received a warning in the output log file. This has happened because each 3D facial norm model contains a considerable number of points (more than twenty thousand points in each shape), and the number of faces is high (600 faces). In such a situation, running the program using multiple nodes could be helpful though this process would be highly complicated.

# Appendix
# Computer Programs

## 1 XML Configuration Files

### 1.1 Deterministic Atlas Model: Data XML File

```
<data-set>
    <subject id="face1">
        <visit id="experiment">
            <filename object_id="face">data/face1.vtk</filename>
        </visit>
    </subject>

    <subject id="face2">
        <visit id="experiment">
            <filename object_id="face">data/face2.vtk</filename>
        </visit>
    </subject>

    <subject id="face3">
        <visit id="experiment">
            <filename object_id="face">data/face3.vtk</filename>
        </visit>
    </subject>

    <subject id="face4">
```

```
        <visit id="experiment">
            <filename object_id="face">data/face4.vtk</filename>
        </visit>
    </subject>

    <subject id="face5">
        <visit id="experiment">
            <filename object_id="face">data/face5.vtk</filename>
        </visit>
    </subject>

    .....

    <subject id="face50">
        <visit id="experiment">
            <filename object_id="face">data/face10.vtk</filename>
        </visit>
    </subject>
<data-set>
```

## 1.2 Geodesic Regression Model: Data XML File

```
<data-set>
    <subject id="norm">
        <visit id="face1">
            <aget>18</age>
<filename object_id="caniofacial">data/caniofacial_face1.vtk</filename>
        </visit>
        <visit id="face2">
            <age>22</age>
<filename object_id="caniofacial">data/caniofacial_face2.vtk</filename>
        </visit>
        <visit id="face3">
            <age>33</age>
<filename object_id="caniofacial">data/caniofacial_face3.vtk</filename>
        </visit>
```

```
        <visit id="face4">
            <age>35</age>
<filename object_id="caniofacial">data/caniofacial_face4.vtk</filename>
        </visit>
         <visit id="face5">
            <age>11</age>
<filename object_id="caniofacial">data/caniofacial_face5.vtk</filename>
        </visit>
    </subject>
</data-set>
```

## 1.3 Deterministic Atlas Model: Model XML File

```
<model>
    <!--<model-type>DeterministicAtlas</model-type>-->
    <model-type>BayesianAtlas</model-type>
    <dimension>3</dimension>

    <template>
        <object id="face">
            <deformable-object-type>Polyline</deformable-object-type>
            <attachment-type>varifold</attachment-type>
            <kernel-width>20</kernel-width>
            <kernel-type>keops</kernel-type>
            <noise-std>1</noise-std>
            <filename>data/face1.vtk</filename>
        </object>
    </template>

    <deformation-parameters>
        <kernel-width>20</kernel-width>
        <kernel-type>keops</kernel-type>
    </deformation-parameters>
</model>
```

## 1.4 Geodesic Regression Model: Model XML File

```
<model>
    <model-type>Regression</model-type>
    <dimension>3</dimension>

    <template>
        <object id="caniofacial">
            <deformable-object-type>Polyline</deformable-object-type>
            <attachment-type>Varifold</attachment-type>
            <noise-std>1</noise-std>
            <kernel-width>20.0</kernel-width>
            <kernel-type>keops</kernel-type>
            <filename>data/template.vtk</filename>
        </object>
    </template>

    <deformation-parameters>
        <kernel-width>25</kernel-width>
        <kernel-type>keops</kernel-type>
        <concentration-of-timepoints>1</concentration-of-timepoints>
    </deformation-parameters>
</model>
```

## 1.5 Deterministic Atlas Model: Parameter Optimization XML File

```
<optimization-parameters>
    <optimization-method-type>GradientAscent</optimization-method-type>
    <use-sobolev-gradient>On</use-sobolev-gradient>
    <freeze-template>On</freeze-template>
    <max-iterations>10</max-iterations>
    <max-line-search-iterations>5</max-line-search-iterations>
</optimization-parameters>
```

## 1.6 Geodesic Regression Model: Parameter Optimization XML File

```
<optimization-parameters>
    <optimization-method-type>GradientAscent</optimization-method-type>
     <max-iterations>30</max-iterations>
```

```
    <save-every-n-iters>10</save-every-n-iters>
    <smoothing-kernel-width>20</smoothing-kernel-width>
</optimization-parameters>
```

### 1.7 Deterministic Atlas model execution:

```
source activate deformetrica

deformetrica estimate model.xml data_set.xml -p
optimization_parameters.xml -v INFO
```

### 1.8 Program running from Roar Super Computer

```
#PBS -l nodes=1:ppn=1:gpus=1
#PBS -l walltime=120:00:00
#PBS -l pmem=10gb
#PBS -A mlr36_d_t_k80_default

# Get started
echo " "
echo "Job started on `hostname` at `date`"
echo " "

# Go to the submission directory
cd $PBS_O_WORKDIR
singularity exec --nv deformetrica_aci_latest.sif bash run.sh

# Run the job itself
nvidia-smi

# Finish up
echo " "
echo "Job Ended at `date`"
echo " "
```

## 2 MATLAB Code: Converting vtk file to obj file

```
clear; path(pathdef);close all
FileName = 'face1TEST.vtk';
fileID = fopen(['C:\Users\Carlos\Desktop\',FileName],'r');

tmpLine = 1;
while tmpLine ~= -1
    tmpLine = fgets(fileID);
    if tmpLine == -1
        break
    end
    if strcmp(tmpLine(1:6),'POINTS')
        tmp = textscan(tmpLine,'%s %f %s');
        numVertices = tmp{2};
        vertices = cell(1,tmp{2});
        break
    end
end
indx = 1;
dontReadAgain = 0;
while tmpLine ~= -1
    tmpLine = fgets(fileID);
    %- this next line checks if we have gone too far
    if strcmp(tmpLine(1:8),'POLYGONS')
        dontReadAgain = 1;
        break
    end

    %- split into individual floats
    tmp = textscan(tmpLine,'%f %f %f %f %f %f %f %f %f %f %f %f ');
    %- convert to matrix
    tmp = cell2mat(tmp);
    %- remove the NaNs
    tmp = tmp(~isnan(tmp));
    for ii = 1: size(tmp,2)/3
        vertices{indx} = [tmp(ii*3-2) tmp(ii*3-1) tmp(ii*3)];
        indx = indx+1;
    end
    %
    if size(tmp,2)/3 ~= 3
        break
    end
```

```
end
%-this next line will contain " 'POLYGONS' #faces #edges "
if dontReadAgain == 1
    tmp = textscan(tmpLine,'%s %f %f');
    numFaces = tmp{2};
    numEdges = tmp{3};
else
    tmpLine = fgets(fileID);
    tmp = textscan(tmpLine,'%s %f %f');
    numFaces = tmp{2};
    numEdges = tmp{3};
end

Faces = cell(1,numFaces);
for jj = 1:numFaces
    tmpLine = fgets(fileID);
    tmp = textscan(tmpLine,'%f %f %f %f');
    Faces{jj} = [tmp{2} tmp{3} tmp{4}];
end
fclose(fileID);

%- Done Reading the vtk file now onto writing the obj file

objFileID = fopen([FileName(1:end-4),'.obj'],'w');
fprintf(objFileID,'#obj file created using Matlab Script\n')
%- print vertices
for i = 1:numVertices
    FormatSpecification = 'v %f %f %f\n';
    fprintf(objFileID,FormatSpecification,vertices{i});
end
%-print faces
for i = 1:numFaces
    FormatSpecification = 'f %i %i %i\n';
    fprintf(objFileID,FormatSpecification,Faces{i}+1);
end
fclose(objFileID);
```

# 3 R Code: Converting obj file to vtk file

```
require(Rvcg)
require(Morpho)
require(RvtkStatismo)
```

```
require(readxl)
require(tidyverse)

dir <- "/home/mka86/Desktop/face/"
file_list <- list.files(dir)

for(i in 1:length(file_list)){

face <- read.table(paste(dir,file_list[i],sep=''),header = TRUE, sep=",")

rof3 <- face3[1:5,] ## omit first 6 rows-unnecessary rows

colnames(face)="name"  ## changing column names

sprof=separate(face,name,c("f","x","y","z"),sep=" ") ## seperate one column to 4

mainface=sprof[sprof$f=="v",] ## extracting v rows from the full data

mnface=mainface[,-1] ## extract first column

options(digits=11)
face <- cbind(as.double(mnface[,1]),as.double(mnface[,2]),as.double(mnface[,3]))

vface <-vcgBallPivoting(face)

vtkMeshWrite(vcgClean(vface,sel=0:6),paste("face",i,sep=''),ascii=T)

}
```

# Bibliography


[1] Falconer, D. S. (1973) "Replicated selection for body weight in mice," *Genetical Research Cambridge*, **22**, pp. 291-321.

[2] Truslove, G. M. (1976) "The effect of selection for body weight on the skeletal variation of the mouse," *Genetical Research Cambridge*, **28**, pp. 1-10.

[3] Johnson, D. R., O'Higgins, P., McAndrew, T. J., Adams, L. M., and Flinn, R. M. (1985) "Measurement of biological shape: a general method applied to mouse vertebrae," *Journal of Embryology and Experimental Morphology*, **90**, pp. 363-377.

[4] Johnson, D. R., O'Higgins, P., and McAndrew, T. J. (1988) "The effect of replicated selection for body weight in mice on vertaebral shape," *Genetical Research Cambridge*, **51**, pp. 129-135.

[5] Mardia, K. V. and Dryden, I. L. (1989b) "The statistical analysis of shape data," *Biometrika*, **76**, pp. 271-282.

[6] Mardia, K. V. and Dryden, I. L. (1994) "The study of morphological variation in the hominid fossil record: biology, landmarks and geometry," *Journal of Anatomy*, **26**, pp. 334-340.

[7] O'Higgins, P. (2000) "Shape averages and their bias," *Advances in Applied Probability*, **197(01)**, pp. 103-120.

[8] O'Higgins, P. and Dryden, I. L. (1993) "Sexual dimorphism in hominoids: further studies of craniofacial shape differences in pan, gorilla, pongo," *Journal of Human Evolution*, **24**, pp. 183-205.

[9] Anderson, C. R. (1997) "Object recognition using statistical shape analysis," *PhD thesis, University of Leeds.*

[10] Hastie, T. and Tibshirani, R. (1994) " Handwritten digit recognition via deformable prototypes," *Technical Report, ATT Bell Laboratories.*

[11] Hastie, T. and Simard, P. Y. (1998) "Metrics and models for handwritten character recognition," *Statistical Science*, **13(1)**, pp.54–65.

[12] Bookstein, F. L. (1996b) "Biometrics, biomathematics and the morphometric synthesis," *Bulletin of Mathematical Biology*, **58**, pp.313–365.

[13] Mardia, K. V., Bookstein, F. L., and Kent, J. T. (2013a) "Alcohol, babies and the death penalty: Saving lives by analysing the shape of the brain," *Significance*, **10(3)**, pp.12-16.

[14] Lohmann, G. P. (1983) "Eigenshape analysis of microfossils: a general morphometric procedure for describing changes in shape," *Mathematical Geology*, **15**, pp.659–672.

[15] Bookstein, F. L. (1986) "ize and shape spaces for landmark data in two dimensions (with discussion)," *Statistical Science*, **1**, pp.181-242.

[16] Kendall, D. G. and Kendall, W. S. (1980) "Alignments in two dimensional random sets of points," *Advances in Applied Probability*, **12**, pp.380-424.

[17] Small, C. G. (1988) "Techniques of shape analysis on sets of points," *International Statistical Review*, **56**, pp.243–257.

[18] Dryden, I. L., Hirst, J. D., and Melville, J. L. (2007) "Statistical analysis of unlabeled point sets: comparing molecules in chemoinformatics," *Biometrics*, **63(1)**, pp.237–251.

[19] Czogiel, I., Dryden, I. L., and Brignell, C. J. (2011). . , 5: 2603–2629. page 13, 349, 375 (1988) "Bayesian matching of unlabeled marked point sets using random fields, with an application to molecular alignment," *Annals of Applied Statistics*, **5**, pp.2603-2629.

[20] Wagener, M., Sadowski, J., and Gasteiger, J. (1995) "Autocorrelation of molecular surface properties for modeling Corticosteroid Binding Globulin and Cytosolic Ah receptor activity by neural networks" *Journal of the American Chemical Society*, **117**, pp.7769-7775.

[21] Horgan, G. W., Creasey, A., and Fenton, B. (1992) " Superimposing two-dimensional gels to study genetic variation in malaria parasites" *Electrophoresis*, **13**, pp.871-875.

[22] O'Higgins, P. and Dryden, I. L. (1992) " Studies of craniofacial development and evolution" *Archaeology and Physical Anthropology in Oceania*, **27**, pp.105–112.

[23] Dryden, I. L. and Mardia, K. V. (1998) "Statistical Shape Analysis," *John Wiley Sons*. ISBN 978-0-471-95816-1.

[24] Ziezold, H. (1994) "Mean Figures and Mean Shapes Applied to Biological Figure and Shape Distributions in the Plane," *Biometrical Journal*, **36**(4), pp. 491-510.

[25] Thompson, D. (1992) "On Growth and Form (Canto) (J. Bonner, Ed.)," *Cambridge: Cambridge University Press*, ISBN 9781107325852.

[26] S. Durrleman, M. Prastawa, N. Charon, J. R. Korenberg, S. Joshi, G. Gerig, and A. Trouv´e (2014) "Morphometry of anatomical shape complexes with dense deformations and sparse parameters," *NeuroImage*, **101**, pp. 35–49.

[27] M. I. Miller, A. Trouv´e, and L. Younes (2006) "Geodesic shooting for computational anatomy," *Journal of mathematical imaging and vision*, **24**(2), pp. 209–228.

[28] Trouvé A. (1998) "Diffeomorphisms groups and pattern matching in image analysis," *Int. J. Comput. Vision* **28**, pp. 213–221.

[29] Beg M. F., Miller M. I., Trouvé A. and Younes L. (2005) "Computing large deformation metric mappings via geodesic flows of diffeomorphisms," *Int. J. Comput. Vision*, **61**, pp. 139–157.

[30] Younes L. (2010) "Shapes and Diffeomorphisms," *Springer*, **171**. ISBN 978-3-662-58495-8.

[31] Alexandre B., Maxime L., Benoît M. and Stanley D. (2018) "Deformetrica 4: an open-source software for statistical shape analysis," *ShapeMI @ MICCAI 2018*, Granada, Spain. ⟨hal-01874752v2⟩.

[32] Kendall, D. G. (1977) "The diffusion of shape," *Advances in Applied Probability*, **9**, pp. 428-430.

[33] Dryden, I.L. and Mardia, K.V. (2016) "Statistical Shape Analysis, with Applications in R," *John Wiley and Sons, Chichester*, Second Edition.

[34] Stegmann, Mikkel B., Gomez , David Delgado (2002) "A Brief Introduction to Statistical Shape Analysis," *Technical University of Denmark, Lyngby.*

[35] Amy Ross (2000) "Procrustes Analysis," *Department of Computer Science and Engineering, University of South Carolina, SC 29208.* https://cse.sc.edu/ songwang/CourseProj/proj2004/ross/ross.pdf

[36] Dryden, I.L. and Mardia, K.V. (1998) "Statistical Shape Analysis," *John Wiley and Sons, Chichester.*

[37] Rao, C. R. (1948) "Tests of significance in multivariate analysis," *Biometrika*, **35**, pp. 58-79.

[38] Reyment, R. A., Blackith, R. E., and Campbell, N. A. (1984) "Multivariate Morphometrics," *Academic Press, New York.*, Second Edition.

[39] Dryden, I. L. and Mardia, K. V. (1977) "Multivariate shape analysis," *Sankhya Series A*, **55**, pp. 460-480.

[40] O'Higgins, P. (1989) "A morphometric study of cranial shape in the Hominoidea," *PhD thesis, University of Leeds.*

[41] Thompson, D. W. (1917) "On Growth and Form," *Cambridge University Press, Cambridge*.

[42] Medawar, P. B. (1944) "The shape of a human being as a function of time," *Royal Society of London, Series B*, **132**, pp. 133-141.

[43] Sneath, P. H. A. (1967) "Trend surface analysis of transformation grids," *Journal of Zoology,London*, **151**, pp. 65-122.

[44] Kendall, D. G. (1983) "The shape of Poisson-Delaunay triangles," *In: Studies in Probability and Related Topics (eds M. C. Demetreseu and M. Iosifescu)*, Nagard, Montreal, pp. 321–330.

[45] Kendall, D. G. (1984) "Shape manifolds, Procrustean metrics and complex projective spaces," *Bulletin of the London Mathematical Society*, **16**, pp. 81-121.

[46] Kendall, D. G. (1985) "Exact distributions for shapes of random triangles in convex sets," *Advances in Applied Probability*, **17**, pp. 308-329.

[47] Kendall, D. G. (1989) " A survey of the statistical theory of shape (with discussion)," *Statistical Science*, **4**, pp. 87-120.

[48] Kendall, D. G. (1991a) "Discussion to 'Procrustes methods in the statistical analysis of shape' by C. R. Goodall," *Journal of the Royal Statistical Society, Series B*, **53**, pp. 321-324.

[49] Kendall, D. G. (1991b) "The Mardia–Dryden distribution for triangles – a stochastic calculus approach," *Journal of Applied Probability*, **28**, pp. 225-230.

[50] Kendall, D. G. (1995) "Looking at geodesics in the shape space for 4 points in 3 dimensions," *In: Current Issues in Statistical Shape Analysis (eds K. V. Mardia and C. A. Gill)*, University of Leeds Press, Leeds. pp. 6-8.

[51] Bookstein, F. L. (1978) "The Measurement of Biological Shape and Shape Change," *Lecture Notes on Biomathematics*, Vol. 24. Springer-Verlag, New York.

[52] Bookstein, F. L. (1984) " A statistical method for biological shape comparisons," *Journal of Theoretical Biology*, **107**, pp. 475-520.

[53] Bookstein, F. L. (1986) " Size and shape spaces for landmark data in two dimensions (with discussion)," *Statistical Science*, **1**, pp. 181-242.

[54] Bookstein, F. L. (1991) " Morphometric Tools for Landmark Data: Geometry and Biology," *Cambridge University Press, Cambridge*.

[55] Bookstein, F. L. (1991) " Morphometric Tools for Landmark Data: Geometry and Biology," *Cambridge University Press, Cambridge*.

[56] Bookstein, F. L. (1997) " Shape and the information in medical images: a decade of the morphometric synthesis," *Computer Vision and Image Understanding*, **66**, pp. 97-118.

[57] Bookstein, F. L. (2000) " reases as local features of deformation grids," *Medical Image Analysis*, **4**, pp. 93-110.

[58] Bookstein, F. L. (2013a) " Allometry for the twenty-first century," *Biological Theory*, **7(1)**, pp. 10-25.

[59] Bookstein, F. L. (1997) " Landmark methods for forms without landmarks: localizing group differences in outline shape," *Medical Image Analysis*, **1(3)**, pp. 225-244.

[60] T. F. Cootes and C. J Taylor (2001, October) " Statistical Models of Appearance for Computer Vision," *Tech. Report, University of Manchester*, http://www.isbe.man.ac.uk/~bim/.

[61] C. Goodall (1991) " Procrustes methods in the statistical analysis of shape," *Royal Statistical Society, Series B*, **53**, pp. 285-339.

[62] GUSTA ME (2014) " Procrustes Analysis," *https://mb3is.megx.net/gustame/other-methods/procrustes-analysis*, accessed January, 2021.

[63] Gower, John C. and Dijksterhuis, Garmt B. (2004) " Procrustes Problems," *Oxford University Press*.

[64] Cairns, Matthew James Francis (2000) " An Investigation into the use of 3D Computer Graphics for Forensic Facial Reconstruction," *Glasgow University*.

[65] Ashburner J., Hutton C., Frackowiak R., Johnsrude I., Price C. and Friston K. (1998) " Identifying global anatomical differences: deformation-based morphometry," *Hum. Brain Mapp.*, **6**, pp. 348-357.

[66] Chung M., Worsley K., Paus T., Cherif C., Collins D. and Giedd J. (2001) " A unified statistical approach to deformation-based morphometry," *NeuroImage*, **14**, pp. 595-606.

[67] Durrleman S., Pennec X., Trouvé A. and Ayache N. (2008) "A forward model to build unbiased atlases from curves and surfaces," in 2nd Medical Image Computing and Computer Assisted Intervention. *Workshop on Mathematical Foundations of Computational Anatomy*, New York, pp. 68–79.

[68] Vaillant M., Miller M. I., Younes L. and Trouvé A. (2004) "Statistics on diffeomorphisms via tangent space representations," *NeuroImage*, **23**, pp. S161–S169.

[69] Fishbaugh, J., Durrleman, S., Prastawa, M., Gerig, G. (2017) "Geodesic shape regression with multiple geometries and sparse parameters," *Medical image analysis*, **39**, pp. 1-17.

[70] Mary L. Marazita, Seth Weinberg, Zachary Raffensperger (2017) "3D Facial Norms Database," *FaceBase Consortium*, retrieved from `https://doi.org/10.25550/1WWP`.

[71] Weinberg, Seth and Raffensperger, Zachary and Kesterke, Matthew and Heike, Carrie and Cunningham, Michael and Hecht, Jacqueline and Kau, Chung and Murray, Jeffrey and Wehby, George and Moreno, Lina and Marazita, Mary (2015) "The 3D Facial Norms Database: Part 1. A Web-Based Craniofacial Anthropometric and Image Repository for the Clinical and Research Community," *The Cleft Palate-Craniofacial Journal*, **53**(6), pp. 185–197.

[72] Brinkley J. F., Fisher S., Harris M. P., Holmes G., Hooper J. E., Jabs E. W., Jones K. L., Kesselman C., Klein O. D., Maas R. L., Marazita M. L., Selleri L., Spritz R. A., van Bakel H., Visel A., Williams T. J. and Wysocka J. (2016) "The FaceBase Consortium: a comprehensive resource for craniofacial researchers," *FaceBase Consortium, Chai Y. Development*, **143**(14), pp. 2677-88.

[73] Cignoni P., Callieri M., Corsini M., Dellepiane M., Ganovelli F.and Ranzuglia G. (2008) "MeshLab: an Open-Source Mesh Processing Tool," *Sixth Eurographics Italian Chapter Conference*, pp. 129-136.

[74] ISTI-CNR (2005) "MeshLab," accessed August 25, 2020, retrieved from www.meshlab.net

[75] James A., Berk G. and Charles L. (2005) "ParaView: An End-User Tool for Large Data Visualization, Visualization Handbook," *Elsevier*, ISBN-13: 978-0123875822.

[76] Utkarsh A. "The ParaView Guide: A Parallel Visualization Application," *Kitware*, ISBN 978-1930934306.

[77] Dai H., Pears N., William S. and Duncan C. (2020) "Statistical Modeling of Craniofacial Shape and Texture," *International Journal of Computer Vision* , **128**, pp. 547-571.

[78] Costa, C., Antonucci, F., Pallottino, F. and Aguzzi, J. (2011) "Shape Analysis of Agricultural Products: A Review of Recent Research Advances and Potential Application to Computer Vision," *Food and Bioprocess Technology*, **4**(5), pp. 673-692.

[79] Wuhrer, S., Shu, C. (2013) "Estimating 3D human shapes from measurements," *Machine Vision and Applications* , **24**, pp. 1133-1147.

[80] Liu, X., Shi, Y., Dinov, I. and Mio, W. (2010) "A Computational Model of Multidimensional Shape," *International Journal of Computer Vision*, **89**, pp. 69-83.

[81] Grosgeorge, D., Petitjean, C., Dacher, J.-N., and Ruan, S. (2013) "Graph cut segmentation with a statistical shape model in cardiac MRI," *Computer Vision and Image Understanding*, **117** (9), pp. 1027-1035.

[82] Tavakoli, V., Amini, A. (2013) "A survey of shaped-based registration and segmentation techniques for cardiac images," *Computer Vision and Image Understanding*, **117** (9), pp. 966-989.

[83] Nakao, M., Nakamura, M., Mizowaki, T., and Matsuda, T. (2021) "Statistical deformation reconstruction using multi-organ shape features for pancreatic cancer localization," *Medical Image Analysis*, **67**: 101829.

[84] Pramanik, R. and Bag, S. (2018) "A survey of shaped-based registration and segmentation techniques for cardiac imagesShape decomposition-based handwritten compound character recognition for Bangla OCR," *Journal of Visual Communication and Image Representation*, **50**, pp. 123-134.

[85] Wazarkar, S. and Keshavamurthy, B. (2018) "A survey on image data analysis through clustering techniques for real world applications," *Journal of Visual Communication and Image Representation*, **55**, pp. 596-626.

[86] Davies, R.H., Twining, C.J., Cootes, T.F., Waterton, J.C. and Taylor, C.J. (2002) "A minimum description length approach to statistical shape modeling," *Medical Imaging, IEEE Transactions on*, **21**, pp. 525-537.

[87] Granger, S. and Pennec, X. (2002) "A minimum description length approach to statistical shape modeling," in *Computer Vision —ECCV 2002*, **2353**, Heyden, A., Sparr, G., Nielsen, M. and Johansen, P. Eds., ed: *Springer Berlin Heidelberg*, 2002, pp. 418-432.

[88] Brechbuhler, C., Gerig, G. and Kubler, O. (1995) "Parametrization of closed surfaces for 3-D shape description," *Computer Vision and Image Understanding*, **61**, pp. 154-170.

[89] Gerig, G., Styner, M., Jones, D., Weinberger, D. and Lieberman, J. (2001) "Shape Analysis of Brain Ventricles Using SPHARM," presented at the Proceedings of the *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA'01), 2001.*

[90] Gerig, G., Styner, M., Jones, D., Weinberger, D. and Lieberman, J. (2001) "Shape versus Size: Improved Understanding of the Morphology of Brain Structures," presented at the Proceedings of the *Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention, 2001.*

[91] Shen, L., Huang, H., Makedon, F. and Saykin, A. (2007) "Efficient Registration of 3D SPHARM Surfaces," presented at the Proceedings of the *Fourth Canadian Conference on Computer and Robot Vision, 2007.*

[92] Su, Z., Wang, Y., Shi, R., Zeng, W., Sun, J., Luo, F. and Gu, X. (2015) "Optimal Mass Transport for Shape Matching and Comparison," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **37** (11), pp. 2246-2259.

[93] Samuels, B. D. et al. (2020) "FaceBase 3: analytical tools and FAIR resources for craniofacial and dental research," *Development*, **147** (18): dev191213.

[94] Kanther, M., Scalici, A., Rashid, A., Miao, K., Deventer, E.V., Fisher, S. (2019) "Initiation and early growth of the skull vault in zebrafish," *Mechanisms of Development*, **160**: 103578.

[95] Potter, A.S., Potter, S.S. (2015) "Molecular Anatomy of Palate Development," *PLoS ONE*, **10** (7): e0132662.

[96] Roosenboom, J., Lee, M.K., Hecht, J.T., Heike, C.L., Wehby, G.L., Christensen, K. et al. (2018) "Mapping genetic variants for cranial vault shape in humans," *PLoS ONE*, **13** (4): e0196148.

[97] Brinkley, J.F., Borromeo, C., Clarkson, M., Cox, T.C., Cunningham, M.J. et al. (2013) "The ontology of craniofacial development and malformation for translational craniofacial research," *American Journal of Medical Genetics Part C*, **163C**, pp. 232-245.

[98] Mauricio (2019) "Deformetrica: Project ID-11135277," *GitLab*, `https://gitlab.com/icm-institute/aramislab/deformetrica.git`, accessed January 2020.