NOTES FOR SEMINAR

Statistical Shape Analysis

- or -

How to Build a Digital Clone Using Simple Modalities

SS 2012

Stefanie Wuhrer

Cluster of Excellence Multimodal Computing and Interaction
Saarland University
## Contents

1 About These Notes .......................... 2

2 Human Shape Correspondence .......... 3
   2.1 Overview of Template Fitting Approaches ................................. 3
   2.2 Initialization ............................................ 4
   2.3 Posture Fitting ........................................... 6
   2.4 Shape Fitting ............................................. 6
   2.5 Discussion of Energy Optimization ..................................... 7
   2.6 Some Results ............................................... 8
   2.7 Further Reading ........................................... 9

3 Statistical Shape Analysis ........... 11
   3.1 Analyzing Shape ........................................... 12
      3.1.1 Principal Component Analysis .................................. 12
      3.1.2 Independent Component Analysis ................................ 13
      3.1.3 Principal Geodesic Analysis .................................. 14
      3.1.4 Wavelet-Based Shape Analysis .................................. 14
   3.2 Analyzing More Than Shape ................ 16
      3.2.1 Analyzing Shape and Texture .................................. 16
      3.2.2 Analyzing Shape and Posture .................................. 16
      3.2.3 Analyzing Multiple Variations with Multilinear Models ........ 17
   3.3 Further Reading ............................... 18

4 Building Digital Clones ............... 19
   4.1 Learning a Correlation ................................. 19
      4.1.1 Linear Mapping ........................................... 20
      4.1.2 Shared Gaussian Process Latent Variable Models ................. 20
      4.1.3 Optimizing the Prediction ................................... 20
   4.2 Some Applications ............................... 21
      4.2.1 Prediction from Motion Capture Data ................................ 21
      4.2.2 Prediction from Measurements .................................. 21
      4.2.3 Prediction from Images ...................................... 22
Chapter 1

About These Notes

The best material model of a cat is another, or preferably the same, cat.

Arturo Rosenblueth (1900-1970)

In everyday life, geometric shapes surround us, and understanding geometric concepts has been a goal of humans for centuries. Recently, it has become common to digitize three-dimensional shapes using multiple modalities, such as 3D laser-range scanners or image-based reconstruction systems. We aim to use these shapes in applications, such as design, entertainment, scientific and medical visualization, and realistic simulation. In order to achieve this goal, the shapes need to be processed and described in an efficient and informative way. These notes focus on learning statistical information about populations of shapes. The human body, head, and face shapes are particularly interesting, since knowing statistical properties of human shapes has many applications, such as automatically computing digital clones, recognition, and ergonomic design.

These notes are intended to give an overview of different techniques that are commonly used to perform statistical shape analysis. In order to use these techniques, the population of shapes first needs to be parameterized. This is a challenging problem in general, and we focus our discussion on computing correspondences between human body shapes. Finally, we discuss how the statistical models can be used for shape inference from simple modalities, i.e. how to build a digital clone. Note that these notes do not (and are not intended to) give an extensive overview of all existing techniques.

Thanks to Timo Bolkart, Alan Brunton, and Augusto Salazar for helpful discussions about the techniques presented in Chapter 3 of these notes.
Chapter 2

Human Shape Correspondence

*The human mind has first to construct forms, independently, before we can find them in things.*

Albert Einstein (1879-1955)

This chapter outlines how to automatically compute correspondences between human shapes in arbitrary posture. If the human shapes are in different postures, we need to solve a non-rigid correspondence problem.

This is a difficult problem. Assume that we aim to correspond two surfaces $S^{(0)}$ and $S^{(1)}$ containing $n^{(0)}$ and $n^{(1)}$ vertices, respectively. If we try all possible combinations, the search space for solving the correspondence problem has size $O\left(n^{(0)} n^{(1)}\right)$. In practice, this problem is further complicated by the acquisition of the data. Common ways to acquire human body shapes include image-based systems and laser-range scanners. These acquisition devices capture noisy and incomplete data. It is usually difficult to filter the noise and fill the holes in the acquired data because the geometry of the data can be complex.

We use prior information on the geometry of the shapes to aid in solving this task. Since we know that all of the acquired shapes represent human body shapes, we can use a template shape to model the approximate shape of the objects. This significantly restricts the search space of the correspondence problem. If we assume that all the shapes are captured in a standard posture, the template consists only of a triangular mesh. If the shapes are captured in various postures, the template consists of a triangular mesh with associated skeleton and rigging weights.

We first give an overview of template fitting approaches. We then discuss how to fit a template model to a human shape in a standard posture (following [57]). Finally, we extend this approach to human shapes in arbitrary postures (following [56]).

2.1 Overview of Template Fitting Approaches

This section gives an overview of how a template model can be used to aid in solving the correspondence task. Assume first that all the models are acquired in a standard posture. This assumption is relevant in practice, since for many anthropometric surveys, subjects are asked to maintain a standard posture.

The task of computing a correspondence between a population of human body shapes $S^{(0)}, \ldots, S^{(m)}$ represented by possibly noisy and incomplete triangular meshes can now be solved by deforming the given template model $T$ to each shape $S^{(i)}$. After this step, we have $m$ deformed versions of $T$ (and hence, we trivially know the correspondences), and the $i$-th deformed version of $T$ is close to the shape $S^{(i)}$. This approach uses a non-rigid iterative closest point (ICP) method and was first proposed by Allen et al. [1]. In the following, this approach is discussed in more detail.

Note that this approach has the advantage of leading to $m$ complete surfaces in correspondence even when the original acquisitions are incomplete. Hence, we do not need to solve the challenging task of filling holes in scan data explicitly.

The general idea of this approach was previously used in image processing. In this setting, a similar approach is called *classical snake approach* and was first introduced by Kass et al. [35]. The classical snake approach aims to find
an object in an image by fitting a contour to edges of said image. To start, the contour contains the object of interest and is relatively close to its boundary. The contour is then deformed by following the image gradient. As in template fitting approaches, a smoothness term is used to regularize the deformation of the contour. The snake approach was extended to three dimensions [27, 28]. Here, the aim is to reconstruct a three-dimensional surface from a set of input images. The surface should project to the silhouettes observed in the images, its texture should match the images, and it should be smooth. This can be formulated as an energy minimization problem.

Approaches that fit in the non-rigid ICP framework need a good initialization. The reason is that these approaches repeatedly compute the current correspondences using closest points in $\mathbb{R}^3$ and stop when the alignment energy no longer changes significantly. When a poor initialization is used, such an approach can reach a local optimum instead of the global optimum. We discuss in the following how to find a good initial alignment. Once we have a good initial alignment, we can fit the posture of the template to the acquired shape, and finally, we can fit the shape of the template to the acquired shape.

### 2.2 Initialization

This section discusses how to find a good initial shape alignment. We use anthropometric landmarks to find a good initial alignment as follows. We manually mark the locations of a set of anthropometric landmarks on the template model $T$. For each acquired human body shape, we assume that the corresponding anthropometric landmark locations are given, which is the case in many anthropometric surveys. After this step, we know a small set of corresponding points on $T$ and on $S^{(i)}$, and we use these points to compute an initial alignment. In the following, we discuss initial alignment procedures for models in standard posture and for models in arbitrary posture.

#### Standard Posture

For human body shapes in standard posture, we consider 71 of the 73 anthropometric landmarks shown in Figure 2.2. We consider all the landmarks except the ones located on the bottom of the back (landmarks 26 and 27), since the locations of these two landmarks possibly have a large average error. We use these 71 markers as known corresponding points to compute the best rigid alignment of $T$ to $S^{(i)}$.

![Model of the Civilian North American and European Surface Anthropometry Resource (CAESAR) database [42] in standing posture with landmarks shown in red. Figure from [57].](image)

Since the human body shape varies from one individual to another, a rigid transformation is not enough to obtain a good initialization for non-rigid ICP. Hence, we additionally deform $T$ to $S^{(i)}$ with a radial basis function that interpolates the known corresponding landmarks. This approach follows the technique proposed by Xi et al. [58]. That is, we compute a radial basis function using the known landmarks $L^{(T)}$ of $T$ and the predicted landmarks $L^{(S)}$ of $S^{(i)}$. Given two sets of $l$ landmarks $L^{(T)}$ and $L^{(S)}$ (here $l = 71$), we aim to compute a set of three radial basis functions, one for each coordinate, such that $l^{(T)}_j = f(l^{(S)}_j)$, $j = 1, \ldots, l$, where $l^{(T)}_j$ is the $j$-th landmark on $T$ and $l^{(S)}_j$ is the $j$-th landmark on $S^{(i)}$. For radial basis functions, the mapping has the property $f(l^{(S)}_j) = \sum_{k=1}^n w_k \Phi_k(l^{(S)}_j)$, where $\Phi_k$ are
a set of \( l \) basis functions and \( w_k \) are weights. We use Gaussian basis functions, which means \( \Phi_k(r) = \exp(-r^2/c^2) \), where \( c \) is a constant (here \( c = 1.0 \)). The approach deforms all of \( T \)'s vertices using the RBFs to obtain a better alignment of \( T \) to \( S^{(i)} \).

**Arbitrary Posture**

For human body shapes in various postures, we consider the fourteen landmarks shown in Figure 2.2. We use fewer landmarks than for a standard posture, since it is hard to find descriptive posture-invariant node properties for many landmarks.

![Figure 2.2](image)

Figure 2.2: *Figure shows a model with landmarks shown in red and structure of the landmark graph. Figure from [56].*

The template model is shown in Figure 2.3. To align the template to a scan in different posture, it does not suffice to use a rigid alignment. Furthermore, since we only know a small number of corresponding landmarks, it is not suitable to use radial basis functions to align the template to the scan. Hence, we use a skeleton-based articulation model of the template. In this model, each bone of the skeleton is deformed rigidly, and each vertex is deformed by a linear combination of the bone transformations. The weights used in the linear combination are called *rigging weights*. Figure 2.3 shows the template model. The left of the figure shows the template with landmarks and the middle of the figure shows the template and the fitted skeleton. The right of the figure shows the rigging weights by assigning a color to every bone of the skeleton and by coloring each vertex of the template with the color of the bone that has the largest influence on this vertex. This skeleton was computed using the technique by Baran and Popović [6].

![Figure 2.3](image)

Figure 2.3: *Template model with landmarks, skeleton, and rigging weights. Figure from [56].*

We use the locations of the predicted landmarks to find an initial skeleton of \( S^{(i)} \) that has the same structure as the skeleton of \( T \). We use each learned landmark as a vertex of the skeleton and we find the remaining vertices of the skeleton using linear combinations of the learned landmark positions. We then compute transformations from the
sketch of $T$ to the skeleton of $S^{(i)}$ as follows. We first order the bones of the skeleton using the tree structure of the skeleton as in a scene graph. We express the transformation of the root using a rigid transformation consisting of a rotation, a scale factor, and a translation vector. The relative transformation of every other bone with respect to its parent is expressed as a rotation. To compute the initial alignment, we express bone rotations using quaternions because this allows us to compute the transformations based on fewer parameters. If we know the relative rotation of a bone $b_j$ with respect to its parent, it is straightforward to compute the transformation matrix $A_j$ of bone $b_j$.

Given the initial skeleton of $S^{(i)}$ and the skeleton of $T$, we compute the transformations that deform the skeleton of $T$ to the skeleton of $S^{(i)}$ as follows. The global scale factor is computed as the average scale factor of the bones of the torso between the two skeletons. The global translation and rotation are computed to align the bone of the torso. Every other rotation is computed based on the relative positions of adjacent bones using quaternions. Once we know the transformation matrices that deform the skeleton of $T$ to the skeleton of $S^{(i)}$, we can use the skinning weights to align $T$ to $S^{(i)}$.

2.3 Posture Fitting

For models in different postures, the initial alignment computed using a skeleton-based deformation of $T$ discussed in the previous section needs to be refined before non-rigid ICP can be used. This is achieved by first fitting the posture of $T$ to the posture of $S^{(i)}$. That is, we aim to refine the deformation matrices that deform the skeleton of $T$, such that the deformation of $T$ is close to $S^{(i)}$.

In the following, let $B^{(T)}$ denote the skeleton of $T$, and let $B^{(S)}$ denote the skeleton of $S^{(i)}$. We proceed in two steps. In a first step, we optimize the location of $B^{(S)}$ using the known landmark positions by minimizing the energy

$$E_{ld} = \sum_{j=0}^{l-1} \left( \sum_{k=0}^{b-1} w_{jk} A_k \hat{I}_j - l_j \right)^2$$

with respect to the transformation parameters (modeled using quaternions), where, $l$ is the number of landmarks (here $l = 14$), $b$ is the number of bones (here $b = 17$), $w_{jk}$ is the weight for the $k$-th bone and the $j$-th landmark of $T$, $A_k$ is the transformation matrix of the $k$-th bone of $B^{(T)}$, and $l_j$ contains the homogeneous coordinates of landmark $l_j^{(T)}$. During this optimization we restrict the scaling, such that the height of the person is between 1.40 m and 2.10 m. Furthermore, we restrict the angle of the rotation of the head, such that the head cannot face backwards. This first optimization aims to compute the transformations that bring the landmarks of $T$ as close as possible to the landmarks of $S^{(i)}$.

In the second step, we consider all vertices of the shapes. That is, we optimize the location of $B^{(S)}$ using all vertex positions by minimizing the energy

$$E_{vn} = \sum_{j=0}^{n-1} \left( \sum_{k=0}^{b} w_{jk} A_k \hat{p}_j^{(T)} - NN^{(S)} \left( \sum_{k=0}^{b} w_{jk} A_k \hat{p}_j^{(T)} \right) \right)^2$$

with respect to transformation parameters (modeled using quaternions). Here, $n$ is the number of vertices of $T$, $\hat{p}_j^{(T)}$ contains the homogeneous coordinates of the $j$-th vertex of $T$, and $NN^{(S)} \left( \sum_{k=0}^{b} w_{jk} A_k \hat{p}_j^{(T)} \right)$ is the nearest neighbor of the transformed vertex $\sum_{k=0}^{b} w_{jk} A_k \hat{p}_j^{(T)}$ in $S^{(i)}$. This optimization aims to compute the transformations that bring the surface of $T$ as close as possible to the surface of $S^{(i)}$.

After these posture fitting steps, we have computed the body shape of $T$ in the posture of scan $S^{(i)}$. We can now fit the body shape of $T$ to the body shape of $S^{(i)}$ using a non-rigid ICP approach.

2.4 Shape Fitting

This section discusses how to align two shapes that are in similar posture, but that have different shape details. To bring two human body shapes into a similar posture, or to align two human body shapes, we use the techniques discussed in
the two previous sections.

To deform a template $T$ to a scan $S^{(i)}$ in similar posture, we use a vertex-based deformation model. When using vertex-based deformation models, we allow each vertex $p_j^{(T)}$ of $T$ to deform using a $3 \times 4$ transformation matrix $A_j$. The goal is to fit $T$ to the scan $S^{(i)}$ while preserving the overall shape of the surface. This is achieved by minimizing the following energy

$$E = w_{\text{data}}E_{\text{data}} + w_{\text{smooth}}E_{\text{smooth}}$$

with respect to the transformations $A_j$, where $w_{\text{data}}$ and $w_{\text{smooth}}$ are non-negative weights, $p_i^{(0)}$ are the homogeneous coordinates of $p_i^{(0)}$, $N(p_i^{(0)})$ is a neighborhood of $p_i^{(0)}$, and $\text{dist}(A_i,A_j)$ measures the distance between the two matrices $A_i$ and $A_j$. Furthermore, $n$ is the number of vertices of $T$ and $NN(S^{(i)})$ is the nearest neighbor of the transformed vertex $A_jp_j^{(T)}$ in $S^{(i)}$.

The goal of this fitting step is to deform the surface of $T$ to the surface of $S^{(i)}$ while maintaining a locally smooth deformation function. Minimizing the energies $E_{\text{ind}}, E_{\text{nn}}$, and $E$ is a difficult task that depends on good initializations of the transformation matrices and on well-chosen nearest neighbors. The following section discusses the details of how these energies can be minimized.

### 2.5 Discussion of Energy Optimization

This section discusses difficulties and possible remedies related to the optimization of the energies $E_{\text{ind}}, E_{\text{nn}}$, and $E$ discussed in the previous sections. We aim to minimize these non-linear energy functions using a quasi-Newton method. Hence, we need to ensure that we are close to a good minimum before we start the minimization.

We achieve this by first aligning the template to the scan and by initializing the transformation matrices $A_j$ as follows. For $E_{\text{ind}}$, we initialize $A_j$ to the transformations that deform the skeleton of $T$ to the initial skeleton of $S^{(i)}$. This ensures that the first deformation of $T$ is close to $S^{(i)}$. After minimizing $E_{\text{ind}}$, we know the transformations $A_j$ that bring the landmarks of $T$ close to the predicted landmarks of $S^{(i)}$. We use these transformation matrices as initialization to optimize $E_{\text{nn}}$ to ensure that the deformed surface of $T$ is close to the surface of $S^{(i)}$. When minimizing $E$, we assume that the surface of $T$ is already close to the surface of $S^{(i)}$. Hence, we initialize $A_j$ to identity in this case.

Recall that we minimize $E_{\text{nn}}$ and $E$ by repeatedly computing the nearest neighbors of the deformed vertices of $T$ in $S^{(i)}$, and by deforming $T$ to be close to these nearest neighbors. This approach relies on the assumptions that points in $T$ correspond to the same intrinsic locations in $T$ as their nearest neighbors in $S^{(i)}$. This is not always true, as shown in Figure 2.4. The figure shows a two-dimensional cross-section of the three-dimensional shapes. The scan is shown using lines and the template is shown using dashed lines. Furthermore, some correspondences are shown using fat dotted lines. The figure shows two points, denoted by $p_j$, that are the nearest neighbor of their intrinsically corresponding points, and of one additional outlier point each. In the left side of the figure the error is a result of a poor alignment. In the right side of the figure the error is a result of missing data in the scan. Other reasons may be noise during data acquisition and outlier points on $S^{(i)}$.

To remedy these problems, the energy terms $E_{\text{nn}}$ and $E$ commonly discard certain pairs of points and their nearest neighbors. To remedy the first problem, a point and its nearest neighbor are discarded if the angle between their outer normal directions is above a threshold. This ensures that a correspondence cannot map to a surface with opposite orientation as shown in the left of Figure 2.4. To remedy the second problem, a point and its nearest neighbor are discarded if the distance between the two points is above a threshold. This thresholding technique also takes care of problems due to noise and outlier points.
It is not straightforward to set the above two thresholds, especially the threshold discarding point pairs that are too far apart. This threshold needs to be large enough to allow for local misalignments, yet small enough to avoid problems. Recall that when minimizing $E_{nn}$, large regions of $T$ are influenced by one transformation, and when minimizing $E$, we consider an energy term that encourages a locally smooth deformation. This ensures that a single erroneous nearest neighbor match among many correct matches has very little influence. We therefore choose not to discard correspondences based on distances of point pairs. We set the normal angle threshold to 110 degrees.

When minimizing $E$ (see Equation 2.1), two energy functionals are considered, namely $E_{data}$ and $E_{smooth}$. If we give a high weight to $E_{data}$, we can align $T$ very closely to $S^{(i)}$. However, the result may not preserve the human shape. If we give a high weight to $E_{smooth}$, the shape of $T$ will be preserved well. However, the alignment of $T$ to $S^{(i)}$ may be poor. It has been shown experimentally that $E$ is best optimized not using a fixed set of weights, but using a weight relaxation scheme. This scheme proceeds by initially giving high weight to $E_{smooth}$, by minimizing $E$, and by iteratively decreasing the weight of $E_{smooth}$. This scheme allows us to repeatedly find alignments that preserve the shape of $T$, and that fit more and more closely to $S^{(i)}$. We initially set $w^0_{data} = 1$ and $w^0_{smooth} = 10$ and we relax $w^t_{smooth}$ as $w^t_{smooth} = 0.5w^{t-1}_{smooth}$ whenever $E$ does not change much.

It is crucial to encourage a locally smooth deformation between $T$ and $S^{(i)}$. We use the energy $E_{smooth}$ for this. Other approaches may be used, such as extended energy functions or deformation graphs. The main advantage of using $E_{smooth}$ is its conceptual simplicity and its efficient evaluation.

In our implementation, we use kd-trees [4] to speed up the nearest neighbor search and we minimize $E_{ind}$, $E_{nn}$, and $E$ using a quasi-Newton approach [38].

### 2.6 Some Results

This section shows some results for standard posture (from [57]) and for varying posture (from [56]).

For human models in standard posture, the quality of the correspondences is visualized in Figure 2.5. We manually applied a texture to the template model (left of Figure 2.5) and transferred the texture using the correspondences obtained with the predicted landmarks. We can see that the texture map is preserved for models with different body shape.

For human models in varying postures, the quality of the correspondences is visualized in Figure 2.6. We manually applied a texture to the template model (left of Figure 2.6) and transferred the texture using the correspondences obtained with our algorithm. We can see that the texture map is preserved for models with different body postures and body shapes. Furthermore, we manually selected a set of feature points on the template model and assigned a unique color to each feature point. The features were then transferred to the bodies shown in Figure 2.6 using the correspondences obtained with our algorithm. Note that the locations of the features are on corresponding anatomical parts of the bodies.
2.7 Further Reading

Computing dense point-to-point correspondences between two possibly deformed surfaces has received considerable attention in recent years [50]. Although many algorithms have been developed to solve the correspondence problem, only few of these algorithms are suitable for statistical model building and shape analysis.

The reason is that when considering the application of statistical model building, we need to solve for the correspondence and for a smooth transformation between a group of shapes. The reason is that solving only for correspondence may result in high levels of noise, which in turn affects the statistical model significantly.

Blanz and Vetter [8] use a set of landmarks to compute the correspondence between pairs of human faces. Allen et al. [1] use a set of landmarks and a template model to deform a template model to human shapes in similar postures, as described in this Chapter. Amberg et al. [2] improve upon this method by formulating the underlying optimization problem using an iterative linear framework that is shown to yield better convergence. Anguelov et al. [3] extend Allen et al.’s approach to work for varying postures. The approach computes the correspondences between a large database of humans and uses the result to build an animated surface model of a moving person. The database contains one subject in multiple postures and the remaining subjects in the standing posture of the CAESAR database. Hasler et al. [30] improve the approach by using fewer landmark positions and by using a database containing many subjects in multiple postures. Pauly et al. [41] compute the correspondences and the transformation between multiple views of a scan for the application of scan completion using a small set of landmark positions.

Recently, several landmark-free approaches have been proposed. Huang et al. [31] proceed by iteratively alternating between a correspondence optimization and a deformation optimization. The approach can be viewed as an
extension of the Iterative Closest Point algorithm (ICP) [7] that is often used to solve the rigid correspondence problem. The method is shown to perform well if the two meshes are initially well aligned. If the alignment is poor, the method fails. Huang et al. show that the obtained correspondences yield visually pleasing shape interpolations. Zhang et al. [61] propose a technique that solves the correspondence problem by finding a small set of features and by choosing the best feature correspondence as the one that minimizes a deformation energy. To improve the efficiency of the algorithm, the tree of all matching features is pruned if the features are too dissimilar. Nonetheless, the algorithm is not as efficient as the algorithm of Huang et al. [31]. Once the feature correspondences are computed, the full correspondence is found by deforming the full mesh based on the feature points. Chang and Zwicker [12] use a reduced deformable model to compute the correspondence and the transformation between two surfaces. Instead of operating on the surface directly, the approach needs to convert the surface into a voxel grid.

Methods that require neither landmark positions nor a set of user-specified input parameters have been proposed for motion capture [11, 25]. The methods assume that the same shape was captured in several gradually changing poses and use this information to learn a deformation model. Li et al. [36] propose an approach to register pairs of range images without using any landmark positions or input parameters. The method is shown to perform well.

In medical imaging, two methods were proposed recently [19, 60] that solve the correspondence problem using a landmark-free approach that operated on the unit sphere. First, each shape is mapped to the unit sphere. Second, the charts on the unit sphere are aligned to minimize an energy. Davies et al. [19] align the charts until the description length is minimized while Yeo et al. [60] align the charts to minimize a diffeomorphic energy. Both approaches require as input manifold meshes with spherical topology. That is, surfaces with holes cannot be taken into consideration.
Chapter 3

Statistical Shape Analysis

An attempt at visualizing the Fourth Dimension: Take a point, stretch it into a line, curl it into a circle, twist it into a sphere, and punch through the sphere.

Albert Einstein (1879-1955)

In this chapter, we consider the problem of analyzing shape variations. The problem we aim to solve is the following. Given a population of shapes in correspondence, we wish to understand the main modes of shape variation within this population. A human observer can best understand the shape variations if they can be visualized. To achieve this, we represent each shape as a vector in a multi-dimensional space $\mathcal{S}$, we perform statistical analysis in $\mathcal{S}$, and we visualize the result. This is only possible if the representation is complete, that is, given a vector in $\mathcal{S}$, we can reconstruct the unique three-dimensional shape corresponding to this vector.

Another goal when analyzing shape variations is dimensionality reduction. Assume that we can acquire 3D scans of the surface of a human body. To obtain a model with a large amount of detail, hundreds of thousands of vertices need to be stored. Assume now that we can acquire an entire population of thousands of people. Storing and processing hundreds of thousands of vertices for each subject is computationally expensive. We therefore want to leverage the fact that all human body shapes are intrinsically similar and use this understanding to derive a shape representation that is low dimensional and that accounts optimally for the shape variability observed in the training data.

Whenever we consider problems that aim to study high dimensional data, we further may have problems with the infamous "curse of dimensionality". This curse describes the fact that data quickly becomes sparse in high dimensional spaces. The reason is explained by the following example. Assume that we sample a one-dimensional line with a certain sample density $s$. In order to sample a $d$-dimensional domain with the same sample density, $s^d$ samples are required. That is, the number of samples required for statistical reasoning increases exponentially in the input dimension. If we know that we do not have enough data to cover the dimensionality of our data representation (for instance if we have only a few thousand scans of subjects but each scan consists of hundreds of thousands of vertices), we can improve the learning of a statistical model by reducing the dimensionality of the input data, for instance using the techniques described below.

Assume that we are given a set of $k$ shapes $S^{(0)}, \ldots, S^{(k-1)}$ that are in correspondence. The simplest complete representation of a shape $S^{(i)}$ with $n$ vertices is the vector $\mathbf{s}_i = [x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_{n-1}, y_{n-1}, z_{n-1}]^T$, where $(x_j, y_j, z_j)$ are the coordinates of the $j$-th vertex of $S^{(i)}$. When using this representation, it is straight forward to convert between the shape and the vector space information. This representation is suitable to analyze shape variations for populations captured in a standard posture. For populations captured in various postures, more sophisticated representations need to be considered.

Section 3.1 discusses how to analyze shape variations for populations captured in a standard posture. Section 3.2 discusses some approaches to analyze shape and posture and shape and texture for human models.
3.1 Analyzing Shape

This section discusses statistical approaches that have been used to analyze the shape variations of a population of shapes captured in a standard posture. Unless mentioned otherwise, the shape $S^{(i)}$ is represented as a vector $[x_0, y_0, z_0, x_1, y_1, z_1, \ldots, x_{n-1}, y_{n-1}, z_{n-1}]^T$, as discussed above. In this case, the dimensionality of the shape space $\mathcal{S}$ is $3n$.

3.1.1 Principal Component Analysis

Principal Component Analysis (PCA) is an approach that aims to reduce the dimensionality of a data set given as a set of vectors in $\mathbb{R}^m$ to $\mathbb{R}^l$ with $l \leq m$ using a linear transformation, while aligning the data along the main modes of variation of the data. PCA is a linear model, and due to its simplicity, it is often used for shape analysis. Using PCA to analyze shape variations and build a prior shape distribution model is also known as active shape model [16].

More specifically, we aim to express a set of $k$ data vectors $\tilde{x}_0, \ldots, \tilde{x}_{k-1}$ as

$$\tilde{x}_i = \left( \sum_{j=0}^{i-1} a_{ij} \tilde{d}_j \right) + \bar{x},$$

(3.1)

where $\tilde{d}_j$ are a set of basis vectors, $a_{ij}$ are scalar weights, and $\bar{x} = \frac{1}{k} \sum_{i=0}^{k-1} \tilde{x}_i$ is the data mean. We can express this way of reducing the dimensionality in matrix form as

$$X^{(\text{cent})} = AD,$$

(3.2)

where $X^{(\text{cent})}$ is a $(k \times m)$ matrix that contains the centered data $\tilde{x}_i - \bar{x}$ as its rows, $A$ is a $(k \times l)$ matrix with entries $a_{ij}$, and $D$ is a $(l \times m)$ matrix that contains the basis vectors $\tilde{d}_j$ as its rows.

Any set of basis vectors with the property that the pseudo-inverse $D^*$ of $D$ exists can be used for dimensionality reduction as follows. We first compute $A$ as $A = X^{(\text{cent})}D^*$, and we use the rows of $A$ as reduced data in $\mathbb{R}^l$. We can compute a data vector $\bar{x}$ in $\mathbb{R}^m$ from its reduced representative $\bar{a}$ in $\mathbb{R}^l$ as $\bar{x} = \bar{a}D + \bar{x}$.

It remains to compute the basis vectors $\tilde{d}_j$ and the weight matrix $A$. PCA computes the basis vectors with the help of an eigen decomposition of the $m \times m$ sample covariance matrix $\Sigma = \frac{1}{n-1} \sum_{i=0}^{n-1} (\tilde{x}_i - \bar{x})(\tilde{x}_i - \bar{x})^T$. The normalization factor $\frac{1}{n-1}$ is important for the statistical properties of PCA. Let $\lambda_0, \ldots, \lambda_{m-1}$ denote the eigenvalues of $\Sigma$ in non-increasing order, and let $\tilde{e}_0, \ldots, \tilde{e}_{m-1}$ denote the corresponding eigenvectors. We choose $\tilde{d}_j = \tilde{e}_j$ for $j = 0, \ldots, l - 1$. It can be shown that the basis vectors computed using PCA are orthogonal and have the property that $\tilde{d}_j$ corresponds to the $j$-th largest direction of data variability. The coordinate axes $\tilde{d}_j$ are called the principal components of the data.

Whenever we reduce the dimensionality of a data set, we lose information about the data. With PCA, we can compute the percentage of information that is lost using the eigenvalues as $\frac{\sum_{j=0}^{m-1} \lambda_j}{\sum_{j=0}^{m-1} \lambda_j} \cdot 100$.

We can use PCA to visualize the shape variations of a population of shapes captured in a standard posture as follows. We represent each shape $S^{(i)}$ as a vector $\tilde{s}_i$ in $\mathbb{R}^{3n}$, and we perform PCA of this data set, which gives a set of vectors $\tilde{a}_i$ in $\mathbb{R}^l$. Recall that the vectors $\tilde{a}_i$ are aligned along the axes of largest variation. Hence, we can visualize the modes of variability of the population of shapes by sampling points, denoted by $\tilde{a}_i$, along the coordinate axes in $\mathbb{R}^l$ and by computing and visualizing the shape vectors $\tilde{s}_i$ as $\tilde{s} = aD + \bar{x}$, where $\tilde{s}$ is the mean shape of the vectors $\tilde{s}_i$. For this visualization, the $j$-th coordinate axis is usually scaled by eigenvalue $\lambda_j$.

This approach has been applied to analyze the shape variations of human body shapes [1] and human head shapes [8, 59]. Figure 3.1 shows the main modes of variation of a population of 500 body shapes that were parameterized as outlined in Chapter 2.

We can see that PCA does not necessarily give semantically meaningful variations. The variations are merely the most significant ones in terms of data variability. Furthermore, PCA is a global method, that is, each principal component potentially influences each vertex coordinate. This is not always desirable. For instance, we may be interested in the shape variability of the arms only. In this case, it is possible to segment the body into several parts and to perform PCA on each part separately. Alternatively, a local shape analysis method, such as a wavelet method, can be used.
Independent Component Analysis (ICA) assumes that the data are generated by overlaying two or more source signals and it aims to describe the data as a linear combination of these unknown source signals. ICA was first introduced by Jutten and Hérault [34] and Comon [15]. The problem was originally motivated by the so-called *cocktail-party problem*. Imagine that you are in a room where two or more people are speaking simultaneously. In this case, the goal is to split the mixed signal you hear into the different sources coming from each speaker, as this will allow you to understand the words that are spoken by the different speakers.

In the case of shape analysis, we assume that the shapes are generated from two or more source vectors, and we wish to split the details of the shape into the different sources. As in the case of the cocktail-party problem, this will allow us to understand the structure of the sources that generate the shapes.

More formally, ICA aims to describe how the data are generated by linearly mixing different sources $\vec{d}_j$. That is, ICA is a linear model, and hence, Equations 3.1 and 3.2 describe the approach. Because of the context of mixing sources, the data vectors are also called *mixture variables* in this context. As in the previous section, the goal is to estimate both $A$ and $D$. The coordinate axes $\vec{d}_j$ are called the *independent components* of the data.

For ICA, we assume that the basis signals $\vec{d}_j$ are statistically independent and have (unknown) non-Gaussian distributions. Surprisingly, these assumptions suffice to obtain a well-defined problem. That is, based on these two assumptions, we can compute $A$ and $D$. However, we cannot determine the variances of the independent components $\vec{d}_j$ because a scalar multiplier in $\vec{d}_j$ could always be canceled out by dividing the corresponding column of $A$ by the same scalar. We therefore can simply assume that each independent component has unit variance. As a result, we also cannot determine the order of the independent components.

Unlike PCA, ICA proceeds by first estimating $A$, and by finding $D$ as $D = A^* X^{(cent)}$, where $A^*$ is the pseudo-inverse.
of $A$. More specifically, let $\bar{y} = \bar{a}^T X^{(cent)}$, where $\bar{a}^*$ is a $(k \times 1)$ column vector. If $\bar{a}^*$ contains as elements the elements of one of the rows of $A^*$, then $\bar{y}$ is one of the sought independent components $\bar{d}_i$.

Assume that all the independent components have the same distributions. In this case, we can use the central limit Theorem to determine $\bar{a}^*$. The central limit Theorem states that under certain conditions, the distribution of a sum of independent random variables tends toward a Gaussian distribution. That is, the sum of two independent random variables has a distribution that is closer to a Gaussian distribution than the distribution of any of the original two random variables.

Define a vector $\bar{z} = A^T \bar{d}^*$. Substitution yields $\bar{y} = \bar{a}^T x^{(cent)} = \bar{a}^T AD = \bar{z}^T D$. Hence, $\bar{y}$ can be expressed as a linear combination of the independent components $\bar{d}_i$ weighted with weights $z_i$. By the central limit Theorem, the distribution of $\bar{y} = \bar{z}^T D$ is more Gaussian than the distribution of any of the $\bar{d}_i$. Hence, the distribution of $\bar{y}$ is least Gaussian when $\bar{y}$ equals one of the $\bar{d}_i$. In this case, $\bar{z}$ contains one entry “one” and ”zero” entries otherwise. This reasoning about non-Gaussianity allows us to find $\bar{a}^*$ as the vector that maximizes the non-Gaussianity of $\bar{a}^T X^{(cent)}$.

Different definitions for non-Gaussianity have been proposed. To maximize non-Gaussianity, heuristic approaches are commonly used. A detailed discussion of non-Gaussianity measures and ways to minimize them is beyond the scope of this chapter.

We can use ICA to visualize the shape variations of a population of shapes in a similar way as was done for PCA. This approach has been applied to analyze the shape variations of human head shapes [59].

### 3.1.3 Principal Geodesic Analysis

Principal component analysis is a powerful tool to study the shape variability of a population. However, since the mean and the covariances are computed using Euclidean distances, PCA is limited to data lying in a Euclidean vector space. This is not necessarily the case of complex shapes that are densely sampled. Principal Geodesic Analysis (PGA) is a statistical method that generalizes PCA to work for shapes that are elements in a nonlinear Riemannian symmetric space [24]. In this space, the shortest path between two shapes is no longer necessarily a straight line, but it may be a curved path. Hence, the distance between shapes in this space cannot be measured using Euclidean distances, but using geodesic distances of the shape space.

With this new way of measuring distances, we can generalize PCA as follows. The mean is no longer the linear average of the data, but the point in shape space that minimizes the sum of squared geodesic distances to each data point. PCA uses a set of basis vectors that span a Euclidean subspace with maximal variability. Since we no longer operate in a space that is intrinsically Euclidean, PGA aims to find a set of basis curves that span a subspace with maximal variability.

PGA has the advantage of finding a mean and variabilities that are intrinsic to the shape space defined by the training data. PGA has only been used for a few shape representations. PGA has been shown to perform well to study hippocampi in medical imaging [24]. The main disadvantage is that this improved shape space comes at the cost of no longer having a linear formulation for the problem.

While PCA can be solved using linear algebra, PGA requires solving multiple non-linear optimization problems, and the way of solving them depends on the shape representation of the data. One way of solving the problem is to repeatedly linearize the shape space under consideration. This way of considering the problem is utilizes tangent space. The tangent space of a shape space is a linearized projection of the shape space in the neighborhood of a point of interest (most often the mean). The Euclidean distance in tangent space is then used to approximate the geodesic distance in shape space. For a more detailed discussion of tangent space, refer to Dryden and Mardia [22].

### 3.1.4 Wavelet-Based Shape Analysis

All of the methods discussed so far aim to analyze global shape variations. Recall that we aim to express each parameterized shape of the training population by the sum of the mean shape and a set of basis transformations that encode the observed shape variability, as formalized in Equation 3.1. The methods considered so far have basis functions that have a global influence; that is, each basis vector potentially influences each vertex coordinate of $\bar{s}_i$. In the following, we discuss a technique that aims to find local shape variations effectively. More precisely, the aim is to compute spatially localized shape variations at different scales. This has applications in medical imaging and in facial shape analysis, where a medical condition or a facial expression may cause large localized variations.
One technique that can be used to find localized shape variations is the use of wavelets. Wavelets describe brief wave-like oscillating functions and are useful in signal processing since they allow to decompose a signal into its low and high frequency components. Davatzikos et al. [17] used wavelets to perform a statistical analysis of contours in images. Nain et al. [39] extended this technique to use wavelets to perform a statistical analysis of three-dimensional shapes. Since then, wavelets have been used in shape analysis for shape segmentation and synthesis [40, 37, 10]. In the following, we outline the main idea of the approach.

The aim is to express each shape vector \( \vec{s}_i \) using Equation 3.1, where each basis vector \( \vec{d}_j \) only affects a certain spatial location at a certain scale. That is, the basis vectors \( \vec{d}_j \) should be sparse. A set of basis vectors that satisfies this requirement are spherical wavelets. These wavelets can be used in general to analyze any signal on a mesh. Here, we use the vertex locations as signal.

To compute the spherical wavelet basis functions and coefficients, each parameterized shape \( S^{(i)} \) is considered at different resolution levels. If we have a population of parameterized shapes, we can obtain finer representations of these shapes by applying the same consistent subdivision rules to all shapes of the population. A subdivision scheme repeatedly inserts new vertices into the meshes in a systematic way. Figure 3.2 shows three levels of subdivision of a surface. In the following, we consider \( \mathcal{L} \) resolution levels. The set of vertices contained in the mesh at resolution level \( j \) are denoted by \( K_j \), and the set of vertices added at resolution level \( j+1 \) are denoted by \( M_j \). That is, \( K_{j+1} = K_j + M_j \).

![Figure 3.2: Three levels of subdivision of a surface mesh.](image)

Each vertex \( \vec{v}_k \) of shape \( \vec{s}_i \) is now expressed as a linear combination of two types of basis functions. The first type of basis function \( \phi_j, (\vec{v}_k) \) is a hat-shaped scaling function defined for resolution level \( j \) at the nodes of \( K_j \). The second type of basis function \( \psi_j, (\vec{v}_k) \) is a wavelet function defined for resolution level \( j \) at the nodes of \( M_j \). Wavelet functions capture the detail of the new resolution level. Note that the wavelet basis functions do not depend on the coordinates of \( \vec{v}_k \), but only on its connectivity in the subdivision mesh and on the vertex coordinates of \( \vec{v}_k \)’s neighbors. The vertex \( \vec{v}_k \) is now expressed as \( \vec{v}_k = \sum_{r \in K_0} \lambda_{0,r} \phi_{0,r}(\vec{v}_k) + \sum_{0 \leq j} \sum_{m \in M_j} \gamma_{j,m} \psi_{j,m}(\vec{v}_k) + \bar{\vec{v}}_k \), where \( \bar{\vec{v}}_k \) is the average vertex position of \( \vec{v}_k \) over all shapes \( \vec{s}_i \) in the training set, and \( \lambda_{0,r} \) and \( \gamma_{j,m} \) are weights called wavelet coefficients that are analogous to the weights \( a_{ij} \) used in Equation 3.1.

Note that with this representation, if we know the mean vertex positions \( \bar{\vec{v}}_k \) and the wavelet coefficients \( \lambda_{0,r} \) and \( \gamma_{j,m} \), we can reconstruct a shape. The representation is designed, such that each wavelet coefficient influences a localized spatial location at a certain scale. We can now learn the mean and variance over the training population of each wavelet coefficient using a one-dimensional maximum likelihood estimate and use this information to find areas of the shape that have large variance. These areas are the ones that are influenced by the wavelet coefficients with large variance measure. Figure 3.3 shows the color-coded local shape variations over a database of human faces in neutral expression.

![Figure 3.3: Visualization of the mean face colour-coded using the standard deviations at each vertex. Figure from [10].](image)
3.2 Analyzing More Than Shape

The previous section gave an overview of how we can analyze the shape of a population acquired in similar posture. In many applications, we are interested in analyzing shape in conjunction with other attributes, such as texture, human body posture, or facial expression. This section gives an overview of techniques that have been proposed to achieve this goal.

3.2.1 Analyzing Shape and Texture

Blanz and Vetter [8] introduced a technique to analyze both the shape and texture information of a population of three-dimensional face scans. In this work, the face shape and the texture information are considered independently. This model makes sense when the texture is acquired in an uncontrolled setup, where lighting conditions may influence the texture information.

To analyze shape and texture, Blanz and Vetter propose to build a first PCA space from vertex coordinates of the registered 3D face shapes and a second one from the corresponding texture information. This results in a representation that stores for each face a PCA weight vector containing the shape information and a PCA weight vector containing the texture information. This model is called *morphable model* and was used to reconstruct textured 3D faces from images. This application is discussed in more detail in Chapter 4.

3.2.2 Analyzing Shape and Posture

When considering human body shapes, we may be interested in both shape and posture variations. Shape variations are interesting to designers, who wish to design garments that fit most subjects of a target population. Posture variations may be interesting to animators, who wish to animate a human character in a realistic way.

Anguelov et al. [3] proposed a technique to analyze both shape and posture of a population of three-dimensional body scans. In this method, shape and posture are analyzed independently, in a similar way as in the previously discussed method by Blanz et al.

To analyze posture variation, a template model of a human body shape in a standard posture is used. The template is equipped with a skeleton that decomposes the body into different body parts. For each triangle of the template, we know which body part is belongs to. Let \( P^{(0)}, P^{(1)}, \ldots, P^{(l)} \) denote the different postures of one subject used to fit the posture variations. Each training posture \( P^{(l)} \) is represented by a set of transformations that when applied to the template mesh deforms said mesh to posture \( P^{(l)} \). More specifically, let \( t_k \) be a triangle of the template mesh. The triangle corresponding to \( t_k \) in \( P^{(l)} \) is expressed as \( R_{il(k)} Q_{ik} t_k \), where \( R_{il(k)} \) is a matrix describing a rigid transformation of the \( l(k) \)-th bone influencing the body part containing \( t_k \) and \( Q_{ik} \) is a \( 4 \times 4 \) transformation matrix describing the stretch of triangle \( t_k \).

Given a template mesh and a parameterized posture \( P^{(l)} \), we can compute \( R_{il(k)} \) and \( Q_{ik} \) to obtain a shape representation. This computation is not straightforward as finding \( Q_{ik} \) for each triangle is an under constrained problem. To remedy this, a smoothness constraint is used to solve for \( Q_{ik} \). Inversely, if we are given a set of transformations \( R_{il(k)} \) and \( Q_{ik} \), we can compute the resulting posture by applying the transformations and by solving a least-squares problem to find the vertex coordinates.

The approach proceeds by using a subject in different postures to learn a statistical model of the posture parameters \( R_{il(k)} \) and \( Q_{ik} \). Next, a database \( S^{(0)}, S^{(1)}, \ldots, S^{(m)} \) of different subjects in similar posture is considered to study the shape variability. Similar to above, triangle \( t_k \) of the template mesh is deformed to its position in \( S^{(i)} \). The deformation now also incorporates a shape deformation \( S_{ik} \). That is, triangle \( t_k \) is deformed as \( R_{il(k)} S_{ik} Q_{ik} t_k \). The template is fitted to each training shape \( S^{(i)} \) by first solving for \( R_{il(k)} \) and \( Q_{ik} \) as outlined above, and by subsequently solving for \( S_{ik} \) (again, smoothness constrains are used to solve this problem). Finally, a shape deformation model is computed by performing PCA over the information in \( S_{ik} \).

This gives a statistical model of body posture and an independent statistical model of body shape. By combining the two models, we can now visualize both body shape and body posture variations. This model is called *SCAPE model* and has been used to reconstruct body shapes from motion capture data and from images. These applications are discussed in more detail in Chapter 4.
The SCAPE model treats body shape and posture independently. Therefore, it does not model the correlation between shape and posture that leads to muscle bulging, for instance. To remedy this problem, Hasler et al. [30] presented an alternative method to analyze shape and posture. In their model, shape and posture are considered to be correlated. The main idea of this approach is the observation that the shape space of a population of human shapes forms a hyper-surface in a high-dimensional space $\mathbb{R}^{3n}$. Commonly, the space is described using PCA, but it is not ideal, since the shape space may very well be curved. Therefore, a more contact and accurate representation may be obtained by looking at other shape representations. One option is to go from $\mathbb{R}^{3n}$ to a shape space $\mathcal{S}$, where the data is linearized. Hasler et al. achieve this by encoding in $\mathcal{S}$ the difference between a triangle $t_k$ in the template model and its corresponding triangle in model $S^{(i)}$. The difference is encoded using rotation and stretch matrices with respect to $t_k$’s three neighboring triangles.

The shape and posture variations can be analyzed jointly by performing PCA of the shape representations in $\mathcal{S}$. This allows us to explore both shape and posture variations using a single PCA space. Furthermore, it allows for realistic muscle bulging as shape and posture are now correlated.

As an aside, this linearized shape space is suitable to encode near-isometric deformations; that is, deformations that preserve distances measured along the surfaces $S^{(i)}$. A shape space that is similar in spirit has been used for near-isometric shape morphing [55]. This shape space is similar in that it also linearizes near-isometric shapes at the cost of a larger shape representation.

### 3.2.3 Analyzing Multiple Variations with Multilinear Models

Finally, we discuss a general technique to analyze multiple properties at once. These properties can be shape, texture, and posture (as above), or other attributes of interest for a specific application. The technique is called multilinear model, and de Lathauwer [20] developed the statistical methods that form the basis of this technique. Vasilescu and Terzopoulos [51] applied the method to analyze 2D images of faces, and Vlasic et al. [52] applied the method to analyze 3D face scans.

This technique uses multilinear algebra to find the main modes of variation. More specifically, the model arranges the data in a tensor. A $n^{th}$ order tensor is a higher order generalization of vectors (1$^{rd}$ order tensors) and matrices (2$^{nd}$ order tensors) to $n$ orders. For a tensor, we can define a set of $n$ mode-spaces as follows. The $i$-th mode space of a $n^{th}$ order tensor with dimensions $d_1 \times d_2 \times \ldots \times d_n$ is the matrix that contains as columns the $d_1 \cdot \ldots \cdot d_{i-1} \cdot d_{i+1} \cdot \ldots \cdot d_n$ vectors of dimension $d_i$. The tensor can be multiplied by a matrix in the $i$-th mode by multiplying the $i$-th mode matrix of the tensor by said matrix. This multiplication is denoted by $\times_i$.

Recall that for a matrix $M$ of dimensions $d_1 \times d_2$, the singular value decomposition (SVD) decomposes $M$ as $M = USV^T$, where $U$ is an orthonormal matrix of dimension $d_1 \times d_1$ that contains the eigenvectors of $MM^T$, $V$ is an orthonormal matrix of dimension $d_2 \times d_2$ that contains the eigenvectors of $M^TM$, and $S$ is a $d_1 \times d_2$ diagonal matrix. The diagonal elements of $S$ are the square roots of the non-zero eigenvalues of $MM^T$. Note that SVD is closely related to PCA. If we use as $M$ the matrix $X^{(cent)}$ of centered data discussed in Section 3.1.1, the SVD can be used to compute the eigenvectors and eigenvalues used for PCA.

We now discuss a way to extend SVD to tensors. The way of "unfolding" tensors into matrices discussed above allows us to express a tensor $\mathcal{F}$ using an $n$-mode SVD. That is, we compute a matrix SVD for each mode of $\mathcal{F}$ independently, and we use the resulting matrices $U_i$ that contain the left singular vectors to compute a core tensor $\mathcal{C}$ as $\mathcal{C} = \mathcal{F} \times_1 U_1^T \times_2 U_2^T \ldots \times_n U_n^T$. This aims to align the data in $\mathcal{F}$ along the axes of the highest variability among each mode space. However, unlike in the case of matrix SVD, this alignment is not necessarily optimal.

We discuss how $n$-mode SVD can be used for shape analysis using an example from Vlasic et al. [52], where the aim is to analyze shape and expression variations of a set of 3D face models. The input to the method is a set of parameterized face models containing $n$ vertices, where each of the $m$ subjects is present in $k$ different expressions. With this information, a $3n \times m \times k$ tensor is built that contains the shape representation of $x$, $y$, and $z$ coordinates in the first mode, different subjects in the second mode, and different expressions in the third mode. This tensor is expressed using $n$-mode SVD as $\mathcal{F} = \mathcal{C} \times_1 U_1 \times_2 U_2 \times_3 U_3$. Since we are only interested in the variations along the second and third mode spaces, we can now compute a multilinear model as $\mathcal{M} = \mathcal{C} \times_1 U_1$. This allows us to express each face using two sets of weights: one for shape and one for expression. This model has been used to reconstruct 3D face shapes from images, and this application is discussed in more detail in Chapter 4.
3.3 Further Reading

Statistical analysis is a large field and this chapter does not aim to give a representative overview of all methods. Instead, we merely aim to give an insight into how statistical analysis can be conducted on a set of shapes.

For more details on PCA, refer to the tutorial by Shlens [47]. PCA can be generalized by allowing functions that are not linear in $\vec{d}_j$ in Equation 3.1. This is called non-linear principal component analysis and was introduced by Der et al. [21]. In this case, principal axes are replaced by principal curves. A detailed discussion of this case is beyond the scope of this chapter.

For more information on ICA, we refer to the detailed tutorial by Hyvärinen and Oja [32], which summarizes theoretical and practical properties of ICA. Furthermore, ICA can be generalized by allowing functions that are not linear in $\vec{d}_j$ in Equation 3.1. This is called non-linear independent component analysis and was introduced by Hyvärinen and Pajunen [33]. A detailed discussion of this case is beyond the scope of this chapter.

In this chapter, we give a general overview of using spherical wavelet methods for shape processing. However, it is beyond the scope of this chapter to give a detailed explanation of how a wavelet representation can be computed. For detailed information on ways to compute a wavelet representation of a shape, refer to Schröder and Sweldens [43, 44].

For more detailed information on shape analysis in general, refer to the books by Dryden and Mardia [22] and Davies et al. [18].
Chapter 4

Building Digital Clones

Cloning, wow. Who would have thought? There should be a list of people who can and cannot clone themselves.

Ted Danson (*1947)

In this chapter, we consider the problem of estimating the 3D shape of a human body or face model from simple input modalities, such as a set of measurements or an input photograph. While it is more and more common to digitize human body and face shapes using laser range scanners or image-based systems, this type of acquisition device is relatively expensive and difficult to use, and it is not available to most people. However, most people have a camera or a tape measure available, and if we can use the data they can acquire as input to estimate their body shape, then a large target population can easily obtain a model of their 3D body shape, or a digital clone.

This clone can then be used in applications such as virtual shopping, where a person wishes to purchase clothing in an online store. The clone can try on the clothing, and this will give a good idea of how the clothing article will fit the customer. Another possible application is virtual gaming, where a person may wish to add her own character to the game.

Predicting the full 3D body or face shape from a set of measurements or few input photographs is an under-constrained problem. That is, the given measurements or photographs impose constraints on the 3D geometry of the object that was acquired, but there are in general many 3D shapes that satisfy these constraints. One option to remedy this problem is to use a learned shape space found using statistical shape analysis (as discussed in Chapter 3) as prior information to further constrain the predicted shape. This option is discussed in detail in this chapter.

In the following, we assume that we learned a shape space of our target population using statistical shape analysis, for instance using PCA. Hence, each shape of the population can now be expressed using a (relatively small) set of weights \( a_j, j = 0, 1, \ldots, l - 1 \). Furthermore, for each shape represented this way, we can use the statistical model to evaluate the probability that this shape is part of the target population. In the following, we use this information to find an estimate of a 3D shape from simple modalities.

First, we discuss some general mathematical techniques to learn a correlation between the input data and the 3D shapes that is used to compute an estimate. Second, we give some concrete examples of how an estimate can be computed from specific simple input modalities.

4.1 Learning a Correlation

Let \( S^{(0)}, \ldots, S^{(k-1)} \) denote the \( k \) shapes used for training in correspondence, and let \( \bar{s}_i \) denote their coordinate representations. Furthermore, let \( \bar{a_i} = [a_{i0}, \ldots, a_{il-1}] \) denote the representation of \( S^{(i)} \) in a shape space learned using statistical shape analysis (e.g. PCA discussed in Chapter 3.1.1).

We are interested in predicting a new shape of the same type as \( S^{(i)} \) from some set of simple input measures, such as a set of measurements or a photograph from a specific view point. Since the training shapes \( S^{(i)} \) are given and in
correspondence, we can automatically process the shapes to find the measurements or photographs of interest for each $S^{(i)}$. This step yields a derived measure $P^{(i)}$ per input shape. For instance, if we are interested in predicting the full human body shape from the silhouette of a frontal photograph of a person, we compute $P^{(i)}$ as the frontal silhouette of the training shape $S^{(i)}$. In the following, we express the data in $P^{(i)}$ as a parameterized vector $\vec{p}_i$. The dimensionality of $\vec{p}_i$ depends on the type of input modality of interest. Note that if the dimensionality of the derived data $\vec{p}_i$ is high, we can reduce it using PCA. Hence, in the following, we assume that $\vec{p}_i$ is a low-dimensional representation of the input data of interest.

The knowledge of both a shape representation $\vec{a}_i$ and a representation of the input data $\vec{p}_i$ allows to learn a correlation between the input data and the 3D shapes. This correlation can then be used directly to estimate a new shape representation $\vec{a}_{new}$ from a measured set of input data $\vec{p}_{new}$.

### 4.1.1 Linear Mapping

The simplest way to correlate the data is using a linear mapping. For shape analysis, this was proposed by Allen et al. [1]. In this case, we use the training data to find the matrix $B$ that optimizes the sum of squared differences of $\vec{a}_i = B\vec{p}_i$. This matrix can be found using the Moore-Penrose pseudoinverse of the matrix $[\vec{p}_0, \ldots, \vec{p}_{k-1}]^T$.

Once $B$ is known, we can use it to compute a new shape representation $\vec{a}_{new}$ from a measured set of input data $\vec{p}_{new}$ as $\vec{a}_{new} = B\vec{p}_{new}$. The learned shape space can then be used to transform $\vec{a}_{new}$ into a coordinate representation, and the 3D shape can be visualized.

### 4.1.2 Shared Gaussian Process Latent Variable Models

The previously discussed linear method assumes a simplistic correlation between the data sets. In the following, we discuss a more sophisticated method that allows to model more complex correlations.

The Gaussian Process Latent Variable Model is effective to model a high dimensional data set lying on a low-dimensional manifold in a probabilistic way. The model automatically extracts a set of low-dimensional latent variables of an object from a high-dimensional observation space. The Shared Gaussian Process Latent Variable Model (SGPLVM) [48] is a variant, which extends to multiple observations that share the same underlying latent structure. This is precisely what we wish to model because both the shape representation $\vec{a}_i$ and the simulated input data $\vec{p}_i$ are derived from the same "structure" $S^{(i)}$.

More formally, we are given $k$ pairs of observations as training data, namely $[(\vec{a}_0, \vec{p}_0), (\vec{a}_1, \vec{p}_1), \ldots, (\vec{a}_{k-1}, \vec{p}_{k-1})]$. SGPLVM computes a set of latent, or "hidden" variables $R = [r_0, r_1, \ldots, r_{k-1}]$ that describe the manifold containing the observations, where $r_i$ controls the pair $(\vec{a}_i, \vec{p}_i)$. The latent variables are chosen, such that the observations in $\vec{a}_i$ and $\vec{p}_i$ are conditionally independent given the latent structure $R$. Once the mapping is known, it can be used to predict a new latent point $r$ given a new observation $\vec{p}_{new}$. The new latent point $r$ is then used to predict the new observation $\vec{a}_{new}$ that corresponds to $\vec{p}_{new}$.

In theory, the latent space can be multi-modal and therefore multiple solutions could be generated by this method. An example case where this could happen is the following. Assume that we wish to predict the full human body shape of a subject from a frontal silhouette. If every acquired subject randomly decided to either face the camera or to turn her back to it, then the latent space might have two solutions: one would correspond to a forward facing subject and a second to a person facing away. Selecting the wrong mode would result in larger reconstruction errors. Therefore, it is advisable to restrict the experimental setup in such a way that this type of ambiguity is removed.

For shape analysis, this approach was used by Chen and Cipolla [13].

### 4.1.3 Optimizing the Prediction

The methods discussed so far learn a correlation between the expected input data and the population of 3D shapes. This correlation is then used to predict a new 3D shape from a new set of input data. This approach has recently become commonly used and it can yield good estimates in restricted scenarios, such as human body shapes in a standardized posture.
When the training data has large variations, such as both shape and posture variations, the approaches described above can be used to obtain a good initial estimate. This estimate can be improved further by solving a problem-specific optimization problem. The optimization aims to find the point $\vec{a}_{\text{new}}$ that best fits the input data $\vec{p}_{\text{new}}$. In our example of finding a full human body shape from a frontal silhouette, we would aim to find the point $\vec{a}_{\text{new}}$ whose silhouette image is as close to the observed silhouette $\vec{p}_{\text{new}}$ as possible.

### 4.2 Some Applications

This section gives some applications of predicting shapes from simple input modalities.

#### 4.2.1 Prediction from Motion Capture Data

We first discuss a technique proposed by Anguelov et al. [3] to predict full 3D human body shapes from motion capture data. Motion capture describes a technique to acquire a human body in motion. Traditionally, motion capture is acquired by fixing a sparse set of markers to a body and by acquiring these markers at a high frame rate. The markers are usually located close to joint positions, which allows to reconstruct a skeletal structure in motion. However, this technique does not allow to acquire the geometry of the body shape in motion. One way to leverage motion capture data to compute animations of human subjects is to predict the human body shape and posture from the given sparse set of markers for each frame.

Anguelov et al. solve this problem by fitting a learned SCAPE model (see Chapter 3.2) to the motion capture data. Here, we wish to find one body shape parameter and a sequence of posture parameters to estimate a motion sequence that follows the acquired motion capture data as closely as possible.

#### 4.2.2 Prediction from Measurements

![Figure 4.1: Full human body shape prediction using a learned linear correlation. Left: 34 measurements used as input. Middle: 3D scan of the subject whose measurements are used as input data. Right: predicted shape color-coded according to signed distance (in mm) to ground truth.](image)

Another common goal is to predict a full 3D body shape from a set of measurements. This has applications in computed-aided design (virtual dressing room) and computer graphics (gaming). In this scenario, it is commonly assumed that the shapes are captured in a standardized posture because traditionally, anthropometric measurements are acquired in a standard posture. Due to this, we assume in the following that the shape space is learned using PCA.

Allen et al. [1] predict human body shapes from measurements using a learned linear correlation between the measurements and the 3D body shapes (see Section 4.1). Chu et al. [14] use a similar approach. However, they do not represent the shapes using a triangular mesh, but using a set of smooth patches.
Figure 4.1 shows a shape prediction that was computed using a learned linear correlation. The left side shows the 34 measurements used as input, the middle shows the 3D scan of a model of the CAESAR database [42] whose measurements are used as input data, and the right shows the predicted shape. Note that the predicted shape is close to the true 3D body shape.

The problem of predicting shapes from measurements has received considerable attention in the past ten years. In the following, we review some of the proposed approaches. Not all of these approaches use statistical models to restrict the shape prediction.

Wang [53] uses a parameterized database of human shapes consisting of feature patches to find a new human shape based on a given set of measurements. Feature patches are initialized to smooth patches that are subsequently refined to capture geometric details of the body shape. While this refinement does not maintain the smoothness of the patches, the final results are visually smooth. The approach finds models in the database with measurements similar to the given set of measurements and computes the new human shape as a linear combination of these models. Wei et al. [54] use a similar approach, but the models are represented using a layer-based representation. These approaches do not learn a correlation between the expected input data and the 3D shapes.

Seo and Megnenat-Thalmann [45] represent the human body as a triangular mesh with an associated skeleton model. The body shapes are analyzed and represented using PCA. The approach then learns a mapping from the set of measurements measured on the training data to the PCA space using an interpolating radial basis function (RBF) with Gaussian kernel. This mapping has more degrees of freedom than the linear mapping discussed above. For more information on radial basis functions, refer to Dryden and Mardia [22]. Hasler et al. [30] apply this approach to a representation that encodes human body shape and posture simultaneously to predict human body shapes from measurements. For more information on the shape representation, see Section 3.2.

Recently, Baek and Lee [5] presented a technique that uses hierarchical clustering to build a statistical model of the training data. This approach proceeds by clustering the training database and by performing a multi-cluster analysis of the training data. To predict a body shape based on a set of input measurements, the approach finds the shape within the learned shape space that best describes the measurements using an optimization of shape parameters.

### 4.2.3 Prediction from Images

![Full human body shape prediction from two input silhouettes. Input silhouettes are shown in red, silhouettes of the reconstructed model in blue and superpositions of both in grey. Left: front view. Middle: side view. Right: predicted shape color-coded according to signed distance (in mm) to the ground truth. Figure from [9].](image)

Finally, we discuss approaches that aim to estimate 3D shapes of a learned population based on a set of input images. Blanz and Vetter [8] proposed the first approach to predict a 3D face shape from a single input image using a statistical learning approach. They use a parameterized database of 3D face scans in neutral expression to learn a morphable model (explained in Chapter 3.2). Given an input image in neutral expression, the learned morphable
model is searched to find the textured shape that best explains the input image using an optimization technique. This successful approach resulted in multiple follow-up works.

Seo et al. [46] and Boisvert et al. [9] estimate a body shape from two images of a human in a fixed posture. Starting from a parameterized database of human meshes in similar poses, the approaches perform PCA of the 3D data and learn a correlation between the 3D shapes and the input images. Given two input images, the learned correlation and the learned PCA space are leveraged to find a set of PCA weights that corresponds to a 3D shape that matches the input images well.

Figure 4.2 shows a shape prediction that was computed from two input silhouettes. The left side shows the front and side view silhouettes. The red silhouettes are used as input to the method. These silhouettes were generated from a 3D scan of a model of the CAESAR database [42]. The blue silhouettes are the silhouettes of the predicted shapes and the superimpositions of both silhouettes are shown in grey. The right side of the figure shows the predicted shape. Note that the predicted shape is close to the true 3D body shape.

Chen and Cipolla [13] aim to estimate the human body shape in a fixed pose based on a single given silhouette. The approach proceeds by learning a SGPLVM mapping between the space of the input images and a PCA space of the 3D shapes. Ek et al. [23] use a similar approach to estimate the pose of a human body based on a given silhouette.

Guan et al. [26] estimate both the shape and pose of a human body shape from a single photograph with a set of markers to be identified by the user. The approach is based on the SCAPE model, which is discussed in Chapter 3.2. When adjusting the PCA weights, the shape is deformed to best match the image in terms of a shape-from-shading energy. Hasler et al. [29] estimate both the shape and pose of a human body from a photograph of a dressed person. This approach requires a manual segmentation of the background and the human in the image.

Brunton et al. [10] use wavelet analysis (see Chapter 3.1.4) to learn a prior distribution for the 3D shape of a human face. They use this information to predict the 3D face shape from two stereo photographs. In this work, all faces are assumed to have a neutral facial expression.

Vlasic et al. [52] predict a 3D face shape from a single photograph. They use a multilinear model to learn a prior distribution for 3D shape changes and for 3D expression changes as discussed in Chapter 3.2.
Bibliography


