

Automatically Creating Design Models from 3D Anthropometry Data

Stefanie Wuhrer
Saarland University
Saarbrücken, Germany
swuhrer@mmci.uni-saarland.de

Chang Shu
National Research Council of Canada
Ottawa, Canada
chang.shu@nrc-cnrc.gc.ca

Prosenjit Bose
Carleton University
Ottawa, Canada
jit@scs.carleton.ca

ABSTRACT

When designing a product that needs to fit the human shape, designers often use a small set of 3D models, called *design models*, either in physical or digital form, as representative shapes to cover the shape variabilities of the population for which the products are designed. Until recently, the process of creating these models has been an art involving manual interaction and empirical guesswork. The availability of the 3D anthropometric databases provides an opportunity to create design models optimally. In this paper, we propose a novel way to use 3D anthropometric databases to generate design models that represent a given population for design applications such as the sizing of garments and gear. We generate the representative shapes by solving a covering problem in a parameter space. Well-known techniques in computational geometry are used to solve this problem. We demonstrate the method using examples in designing glasses and helmets.

1 Introduction

In designing wearable products, such as garments or head gears, one of the most important objectives is to make the products to fit the humans comfortably. In order to accommodate the human shape variabilities, different sizes are usually created. Traditionally, this is achieved by considering a few key dimensions of the human body. To design a sizing system, anthropometric measurement data of these dimensions are collected and tabulated. Then, a gridding of the dimensions is formed to create the sizes [1] [2].

Although traditional anthropometry has a long history and has accumulated a vast amount of data, it is limited by its tools (mainly tape measure and caliper); the sparse dimensional measurements do not provide sufficient shape information. In many applications, designers need full 3D models as representative shapes for the sizes [2]. These models are sometimes called manikins when the full body is concerned, or head forms in case of the head and face are concerned. For the purpose of generality, we call them *design models* in this paper. These 3D models provide the overall shape of the human body. They need to be chosen carefully to ensure that the manufactured items fit the target population.

Without 3D information, design models are often created by artists who sculpt out the 3D forms, interpolating the measurement dimensions using their experience in creating human shapes [3]. These approaches are labour-intensive and do not create accurate human models.

3D anthropometric data, obtained using 3D imaging technologies, provide detailed shape information. In addition, traditional measurements can also be extracted from the 3D models. Therefore, 3D anthropometric data offers an opportunity to improve the quality of the design models and, at the same time, maintain the simplicity of the traditional design schemes where key body dimensions are used.

Despite the fact that 3D anthropometric databases have been available for two decades, surprisingly little research has been reported concerning using the 3D data to improve the design models. Robinette [4], Ball [5, Chapter 7], and Meunier et al. [3] are recent attempts in this direction. However, all of these methods rely heavily on manual interaction and do not give a systematic way to optimize the design models such that the sizing system provide maximal accommodation.

Furthermore, these models operate based on simple linear statistical models. The use of statistical models assumes that a large database is available that represents the target population well because statistical models need a large sample size to accurately describe the population. However, the acquisition of such a database is costly and time intensive (the acquisition of the Civilian American and European Surface Anthropometry Resource took 4 years and cost \$6 Million¹). In this work, we are interested in computing design models for a specific target population. This population may be small, as in the case of military design applications. For this reason, we operate using the design parameters directly instead of computing a statistical model of the design parameter space. Once we found the best sizes in parameter space, we use the available 3D anthropometric data to compute the 3D design models, which allows us to create realistic 3D shapes. To compute a 3D design model, we use a statistical mean shape of all the anthropometric data corresponding to points represented by that design model.

In this paper, we present a method for automatically creating a set of design models based on a 3D anthropometric database and optimally computing the design models that represent the underlying target population. Starting with a 3D anthropometric database X that represents the target population for the product to be designed, the method requires as input the ordered set of d measurements M on the human body that need to be considered during the design phase. Furthermore, the method requires as input an ordered set of d tolerances T ; one tolerance for each measurement in M .

For instance, when designing glasses, the set M to be considered are usually the width of the face and the width of the bridge of the nose. In this example, we may know that the glasses to be designed can be adjusted by $2.76cm$ to fit faces with different widths and by $0.19cm$ to fit noses with different widths of the bridge. In this case, T contains the tolerances 2.76 and 0.19 . If the product is non-adjustable, the tolerances represent the measurement intervals that the sizes cover. With this information, the method finds a set of design models that optimally represent the population acquired in the database for the specific purpose of designing a product that depends on the measurements M and that allows adjustments along the tolerances T .

The choice of M and T may have a significant influence on the shape of the design models. Designers, from their experience, usually know the important parameters to consider. These parameters are also determined by the specific requirements of the design. Our goal here is to provide a tool for computing the optimal design models for a given set of design parameters. Designers may use the tool to experiment with different parameters.

Our method proceeds by mapping each of the subjects in the database to a point in \mathbb{R}^d and by solving a discrete covering problem in this space. We use the subjects in the database directly instead of learning their underlying probability distribution since the type of distribution is usually unknown, and since accurate statistical models require large databases. The design models created using this approach are expected to have higher data accuracy and completeness than traditionally produced design models because we compute the best design models from a database that represents the target population.

Sections 3, 4, and 5 discuss how to convert the problem to a discrete covering problem and how to solve the covering problem. To obtain the design models, the method employs the full Procrustes mean [6] as outlined in Section 6. The design models represent the given database. If the database is large, we expect good design models. If the database contains few subjects, it is possible to assume and fit a probability distribution satisfied by the population and to extrapolate subjects from the database based on that distribution. In case of a small database, this approach may provide more accurate results. Section 7 discusses this option. Section 8 gives design examples.

Some prior work has focused on operating in a parameter space based on traditional sparse anthropometric measurements. If only sparse anthropometric measurements are known, it is not straight forward to find 3D design models because computing a 3D model based on a small set of measurements is an under constrained problem. In this work, the shape information is provided by a 3D anthropometric database. This data allows to compute design models by constraining them to lie within the shape space spanned by the database.

In this work, we use the Civilian American and European Surface Anthropometry Resource (CAESAR) database [7] to compute a sizing system. This database contains human shapes in a similar posture. Furthermore, each shape contains a set of 73 anthropometric landmarks. We exploit this information to parameterize the models.

2 Related Work

McCulloch et al. [8] used the traditional sparse anthropometric measurements to automatically create an apparel sizing system that has good fit. For a fixed set of measurements on the body, a fixed number of sizes, and a fixed percentage of the population that needs to be fitted by the sizing system, they optimize the fit of the sizing system. The fit for a human body is defined by a weighted distance function between the measurements on the body and the measurements used for the sizing system. Optimizing the fit amounts to solving a non-linear optimization system. This system is hard to solve and suboptimal solutions may be obtained. This approach can operate in multi-dimensional spaces. It is not straight forward to obtain design models from the resulting sizing system because computing a design model from a sparse set of measurements is an under constrained problem.

¹<http://www.sae.org/standardsdev/tsb/cooperative/caesumm.htm>

Mochimaru and Kouchi [9] represented each model in a database of human shapes using a set of manually placed landmark positions. They proposed an approach to find representative three-dimensional body shapes. The approach first reduces the dimensionality of the data using multi-dimensional scaling, and then uses Principal Component Analysis (PCA) to find representative shapes. Mochimaru and Kouchi showed that this approach is suitable to find representative shapes of a human foot. While this approach is fully automatic, it assumes that the distortion introduced by multi-dimensional scaling is small. While this may be true for low-dimensional data, it is not in general true when three-dimensional measurements of high resolution are considered. Hence, there is no guarantee that the design models optimally represent the population.

Recently, three-dimensional measurements of high resolution have been used to aid in the design process. Robinette [4] studied different ways to align a database of 3D laser scans of heads for helmet design. The goal is to find alignments with minimum variability of shape. This way, one can design helmets that offer maximum protection. Meunier et al. [10] used a database of 3D laser scans of heads to objectively assess the fit of a given helmet. This assessment strategy is useful to analyze existing designs, but it does not produce design models. Kouchi and Mochimaru [11] proposed an approach to design spectacle frames based on database of human face shapes. The approach proceeds by analyzing the morphological similarities of the faces in the database and by dividing the faces into four groups. For each group, a representative form is found automatically. In this work, a spectacle frame was designed for each representative shape, and it was shown that a good fit was achieved. Guo et al. [12] proposed a similar approach to construct design models for helmet design using a database of magnetic resonance images of heads. The approach proceeds by automatically dividing the head shapes into groups and by constructing one design model per group. Although the methods by Kouchi and Mochimaru and Guo et al. require little manual work, the methods are specific to spectacle and helmet design, respectively, and cannot easily be extended to the design of other gear or garments. Furthermore, the division of heads into groups is not guaranteed to produce design models that optimally represent a target population.

Meunier et al. [3] propose to parameterize a set of head scans and to perform PCA on the parameterized scans. They then use the first two principal components to manually design a set of three design models. While this approach aims to exploit the information provided by a parameterized database of head shapes, the approach is heuristic and assumes that the data follows a Gaussian distribution. Furthermore, manually picking the design models based on a learned distribution is hard in a high-dimensional space without reducing the dimensionality of the data because humans cannot easily visualize high-dimensional spaces. Hence, if the aim is to consider four or more principal components, the approach by Meunier et al. becomes very difficult to use.

We propose a fully-automatic method to compute design models that represent a given 3D anthropometric database well. Since the approach computes the design models automatically, it can operate in high-dimensional spaces. Unlike the method by McCulloch et al. [8], our method does not rely on solving a non-linear optimization system. Instead, we model the fit explicitly using a set of tolerances (one tolerance along each dimension) that explains by how much the garment or gear can be adjusted along each dimension. These tolerances depend directly on the design and the materials that are used in a specific application. In this way, our method finds optimal design models for a specific task such as helmet design. To our knowledge, this is the first method that simultaneously considers the optimal fit accommodation and design models.

3 Converting 3D Shape Data Into Parameter Space Points

The proposed approach proceeds by first parameterizing all of the n subjects present in the database X . Parameterizing the database refers to the process of re-meshing the models in X , such that anatomically corresponding vertices and triangles obtain the same index. This step is important, since we wish to compare different models in X , and since we can only compare two models if we know anatomically corresponding parts. In general, finding accurate point-to-point correspondences between a set of shapes is a hard problem [13]. However, in our application, we know that the shapes are human body shapes, which allows the use of template-based approaches to parameterize the database.

First, consider the case where the scans are assumed to be in similar posture. This assumption is common in our application because in a typical 3D anthropometry survey, the human subjects are asked to maintain a standard posture. In this case, we can proceed by first using Ben Azouz et al.'s [14] approach to automatically predict landmarks on the scans followed by Xi et al.'s approach [15] to parameterize the models. Ben Azouz et al.'s approach proceeds by learning the locations of a set of anthropometric landmarks from a database of human models using a Markov Random Field (MRF), and by using probabilistic inference on the learned MRF to automatically predict the landmarks on a newly available scan. Xi et al.'s approach [15] exploits the anthropometric landmarks to fit a template model to the scans. The method proceeds in two steps. First, it computes a radial basis function that maps the anthropometric landmarks on the template mesh to the corresponding landmarks on the scan. This function is used to deform the template mesh. Second, the method further deforms the template to fit the scan using a fine fitting approach as in Allen et al. [16].

Second, consider the general case where the postures of the scans vary. In this case, we can again proceed by first predicting landmarks on the scans automatically and by using a template-based deformation to parameterize the models. When using this method [17], the locations of the landmarks are learned and predicted in an isometry-invariant space and the template fitting approach uses a skeleton-based deformation to allow for posture variation.

In this work, we use the CAESAR database. The models of this database are in similar posture and for each model, we know a set of 73 manually placed anthropometric landmarks. Hence, we use the anthropometric landmarks as input to Xi et al.'s approach [15] to parameterize the models.

Once the database is parameterized, each subject X_i of the database is represented by a triangular mesh, where anatomically corresponding vertices and triangles have the same index, respectively. Hence, we can now measure corresponding length measurements on all of the bodies. This paper considers Euclidean distances. However, the techniques outlined in the following extend to arbitrary length measurements. For instance, the techniques could also be used for parameter spaces that contain geodesic distances or lengths of feature curves on the models.

The approach computes the ordered set M of d distances that are meaningful for the design of a specific garment or gear for each of the n bodies in the database. For each body, the d measurements in order can be viewed as a point P_i in the parameter space $\mathcal{S} = \mathbb{R}^d$. The entire database is then represented as a set P of n points P_0, P_1, \dots, P_{n-1} in \mathcal{S} .

The designer can specify a range of fit along each of the d dimensions that were measured on the bodies. This range specifies how much a garment stretches or by how much gear can be adjusted in the given direction. The ordered set T of d ranges in order defines the side lengths of a d -dimensional box in \mathcal{S} . Let $s_i(B)$ denote the side length in dimension i and let B denote the d -dimensional box with side lengths $s_i(B)$ centered at the origin.

The problem of computing a sizing system that fits the given population can now be expressed as a covering problem in \mathcal{S} . That is, we aim to cover all of the points in P with translated copies of B . Sections 4 and 5 discuss how to solve this problem.

Once the points in P are covered by a set of translated copies of B , we aim to convert each box B_i back into a body shape that represents the points covered by B_i . This is achieved by computing the full Procrustes mean shape [6] of the body shapes corresponding to the points covered by B_i . Section 6 discusses this step in detail.

4 Covering Parameter Space With Boxes

This section discusses the problem of finding the minimum number of translated copies of B that cover all of the points in P .

We first analyze how many translated copies of B need to be considered by the algorithm. When considering only one distance dimension, the box B becomes a line segment of fixed length. Note that two line segments that are combinatorially equivalent (i.e. they cover the same points of P) do not add anything to the space of solutions. Therefore, the algorithm only needs to consider translated line segments that are combinatorially different from each other. Since the set covered by a translated copy of B only changes if a point of P_i either enters or leaves the set, the algorithm only needs to consider the translated line segments with endpoints P_i . Since a line segment has two endpoints and since there are n points in P , we need to consider $2n$ boxes. Extending this argument to d distance dimensions yields that at most $(2n)^d$ translated copies of B need to be considered.

The problem of finding the minimum number of boxes among the set of at most $(2n)^d$ translated copies of B that cover all of the points in P can now be viewed as a set cover problem. A history of this problem can be found in Vazirani [18, Chapter 2]. This problem is NP-hard [19] and it is therefore impractical to find an optimal solution. Hence, we aim to find an approximation to the solution that has bounded error. A solution is a $(1 + \epsilon)$ -approximation of the optimal solution if it uses k' boxes with $k' \leq (1 + \epsilon)k$ for any $\epsilon > 0$, where k is the number of boxes used by the optimal solution. In the following, we discuss how to compute a $(1 + \epsilon)$ -approximation [20]. We then discuss a more efficient greedy algorithm to find a set of boxes that is a $(2d \log(n))$ -approximation of the optimal solution.

4.1 $(1 + \epsilon)$ -Approximation

This section summarizes the approach by Hochbaum and Maass [20] to find a $(1 + \epsilon)$ -approximation to the covering problem. The approach proceeds by dividing the parameter space into a regular grid with width $s_i(B)$ along each dimension. For a given input parameter l related to ϵ ($(1 + \epsilon) = (1 + \frac{1}{l})^d$), the approach partitions the parameter space into slabs of l grid cells along each dimension. Note that l^d partitions are possible because the locations of the slabs can be shifted l times along each dimension. For each of the partitions, the algorithm computes the union of optimal solutions in the sets of $\underbrace{l \times l \times \dots \times l}_{d \text{ times}}$

grid cells created by the partition. The algorithm takes $O(l^d n^{2^d + 1})$ time. For a detailed analysis, refer to [20]. This is only practical for problems with few points in low dimensions. Hence, we only implemented this approach for $d = 2$.

4.2 Greedy Covering

This section summarizes an efficient algorithm to find a solution to the covering problem. Unfortunately, the solution is not guaranteed to be a $(1 + \epsilon)$ -approximation of the optimal solution. The approach proceeds by considering r translated copies of B . We denote these boxes B_0, B_1, \dots, B_{r-1} . For each box B_i , we compute the set of points of P covered by B_i . This

yields a collection of r sets. Our goal is to find a small subset of the sets that covers all of the points in P . We solve the problem using a greedy approach by repeatedly selecting the set covering the maximum number of uncovered points until all points are covered. Finding the set of points covered by one box B_i takes $O(dn)$ time. Hence, finding all of the sets takes $O(dnr)$ time. We store the following information. For each point P_i , we store which boxes contain P_i , for each box B_i , we store which points are contained in B_i , and we store for each box B_i the number of uncovered points in B_i . The set that covers the maximum number of uncovered points can now be found in $O(r)$ time. Furthermore, using the stored information, we can remove one point from all of the boxes B_i in $O(r)$ time. Since there are a total of n points, the algorithm takes $O(nr)$ time. Hence, this approach takes $O(dnr)$ time.

As discussed above, if we consider all of the combinatorially different boxes B_i , r is at most $(2n)^d$. When all combinatorially different boxes are considered, it can be proven that this greedy approach is a $\log(n)$ -approximation of the optimal solution [18, Chapter 2]. Hence, we can find a $\log(n)$ -approximation of the optimal solution in $O(dn^{d+1})$ time.

Denote the set of all combinatorially different boxes by \mathcal{B}^* . In practice, we reduce the number of boxes further by considering only the translated copies of B with centers in P . Denote the set of these boxes by \mathcal{B} . This way, we obtain $r = n$ and reduce the running time to $O(dn^2)$ at the cost of not considering all of the combinatorially different boxes. To analyze the approximation ratio of this solution, consider the case $d = 1$, where the boxes become line segments. Assume that the optimal solution picks a line segment B_i^* not included in \mathcal{B} . Recall that B_i^* is in the set \mathcal{B}^* . All of the elements covered by B_i^* can be covered by two line segments in \mathcal{B} ; namely, the line segment centered at the leftmost point in B_i^* and the line segment centered at the rightmost point in B_i^* . This argument can be generalized to d -dimensional space as follows. A box B_i^* picked by the optimal solution that is not included in \mathcal{B} can be covered by $2d$ boxes in \mathcal{B} . Hence, the optimal solution using \mathcal{B} needs at most $2d$ times the number of boxes picked by the optimal solution using \mathcal{B}^* . Since we solve the problem using the greedy algorithm on \mathcal{B} and since the greedy algorithm is known to compute a $\log(n)$ -approximation of the optimal solution [18, Chapter 2], our approach is guaranteed to compute a $(2d \log(n))$ -approximation of the optimal solution.

5 Covering With Fixed Number of Boxes

In some applications such as the design of garments, it is not desirable to produce sizes for an entire population. Instead, one wishes to produce a fixed number k of sizes that fits the largest portion of the population. For instance, a company that manufactures T-Shirts may wish to manufacture three sizes (small, medium, and large) in a way that the sizes fit the largest possible portion of the population.

In parameter space, this means that we do not wish to cover all of the points in P . Instead, the goal is to cover the maximum number of points in P with a fixed number k of boxes. This problem is also NP-hard since a polynomial-time algorithm to solve this problem would give a polynomial-time algorithm to solve problem considered in Section 4. Hence, we solve this problem using a greedy approach. Unfortunately, the solution is not guaranteed to be a $(1 + \epsilon)$ -approximation of the optimal solution. However, by a similar argument to the one in the previous section, we can show that the approach computes a $(2d \log(n))$ -approximation of the optimal solution. The greedy approach proceeds as in Section 4.2. The only difference is that we stop after the first k boxes are selected. Using an analysis similar to the one in Section 4.2, it can be shown that this algorithm takes $O(dnr)$ time, where r is the number of boxes considered by the algorithm. Recall that $r = O((2n)^d)$ in theory and that we set $r = n$ in practice.

6 Computing Representative Shapes for Boxes

Once the algorithm selected a set of k boxes B_0, B_1, \dots, B_{k-1} to represent the given measurements, we aim to convert each box B_i back into a body shape that represents the points covered by B_i . This is achieved by finding the parameterized body shapes corresponding to points in P covered by B_i and by computing the full Procrustes mean of these shapes [6]. To compute the full Procrustes mean of m shapes, we repeatedly compute the average of the m shapes and align each of the m shapes to the average shape using a rigid transformation.

If the box B_i contains a sufficient number of points and if the mean of these points is close to the center of B_i , the Procrustes mean is a good representative of B_i . Otherwise, the Procrustes mean of these shapes will not yield a good representation of the shapes covered by B_i . If this situation occurs, the approach can be modified by sampling a set of points P_{new} in B_i and by finding the shapes X_{new} corresponding to these points as outlined in Section 7. We can then find a design model by computing the full Procrustes mean of the shapes X_{new} .

This approach yields a set of k body shapes S_i corresponding to the boxes B_i . The shapes S_i can now be used for the design of garments or gear. Note that the shapes S_i are not simply scaled versions of each other since each shape S_i is derived from a different set of true body scans. We use each shape S_i as a design model.

7 Extrapolating Shapes from the Database

The design models computed in the previous sections represent the given database. If the database contains few subjects, the coverage of the target population by the computed design models may be small. In this case, extrapolating subjects from the database may provide more accurate results. In order to extrapolate subjects from the database, we need to assume that the target population obeys a specific probability density. In this section, we assume that the data follows a Gaussian distribution.

We discuss how to extrapolate subjects from the database to improve the coverage of our method. We use the approach by Allen et al. [16] to compute a new shape X_{new} based on a new point P_{new} in \mathcal{S} . Since the database is parameterized and follows a Gaussian distribution, we can perform PCA of the data. In PCA space, each shape X_i is represented by a vector W_i of PCA weights. PCA yields a mean shape μ and a matrix A that can be used to compute a new shape X_{new} based on a new vector of PCA weights W_{new} as $X_{new} = AW_{new} + \mu$. Recall that we aim to create a new shape based on a new point P_{new} in \mathcal{S} . To achieve this goal, we use the database to learn a linear mapping from P_i to $W_i, i = 0, \dots, n - 1$. This mapping is called *feature analysis* and described in detail by Allen et al. [16]. Feature analysis yields a matrix B that can be used to compute a new vector of PCA weights W_{new} based on a new point P_{new} as $W_{new} = BP_{new}$. This model allows to compute a shape X_{new} based on a new point P_{new} as $X_{new} = ABP_{new} + \mu$.

It remains to outline how to find new points P_{new} . We assume that the points $P_i, i = 0, \dots, n - 1$ can be modeled by a Gaussian distribution. We learn the distribution from the given database using maximum likelihood estimation. We then sample a set of points from this distribution and find the corresponding shapes using feature analysis. Figure 1 shows three faces that were obtained using this approach for a training database of 50 faces from the CAESAR database [7].

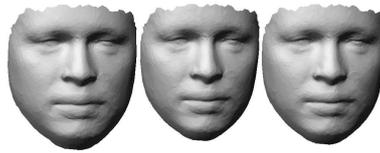


Fig. 1. Faces obtained using feature analysis.

Another way to find new points P_{new} is to sample the surface of a fixed equal probability density of the learned Gaussian distribution (this surface is an ellipsoid). This approach is useful to increase boundary coverage.

8 Examples

We demonstrate the proposed approach for the design of glasses and for the design of helmets. The first example shows the concept of the presented approaches using a small database of faces. The second example gives an evaluation of the greedy approaches using a large database of heads. For the second experiment, we evaluate the quality of the fit of the computed design models using the body dimensions.

For both examples, we set the tolerances based on statistical learning as follows. We compute the variance along each dimension separately, and we set the tolerance along a dimension to four times the variance along said dimension.

8.1 Design of glasses

When designing glasses, one may wish to measure the width of the face and the width of the bridge of the nose. Using these two measurements yields $\mathcal{S} = \mathbb{R}^2$. Figure 2 shows the measurements on one face in the database. The first dimension measures the Euclidean distance between the blue points and the second dimension measures the Euclidean distance between the red points.

For this example, we use 50 faces from the CAESAR database [7]. We choose a small database since this allows to illustrate the result of the $(1 + \epsilon)$ -approximation.

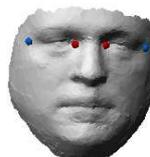


Fig. 2. Measurements used to define parameter space.

In our example, we aim to design glasses that can be adjusted by 2.67cm in the first dimension and by 0.19cm in the second dimension. This defines the box B .

Figure 3 shows the result when covering P using a $(1 + \epsilon)$ -approximation with $\epsilon = 1.25$. The figure shows the points P as black points and the boxes B_i as red boxes. Furthermore, for each box, the figure shows a screen shot of the corresponding Procrustes mean shape. Note that the different face shapes are not merely scaled versions of each other. Instead, they all have different overall shape (height versus width) and they differ in the areas of the jaw, cheeks, eyebrows, and noses. These face shapes can be used to create the sizes for the glasses.

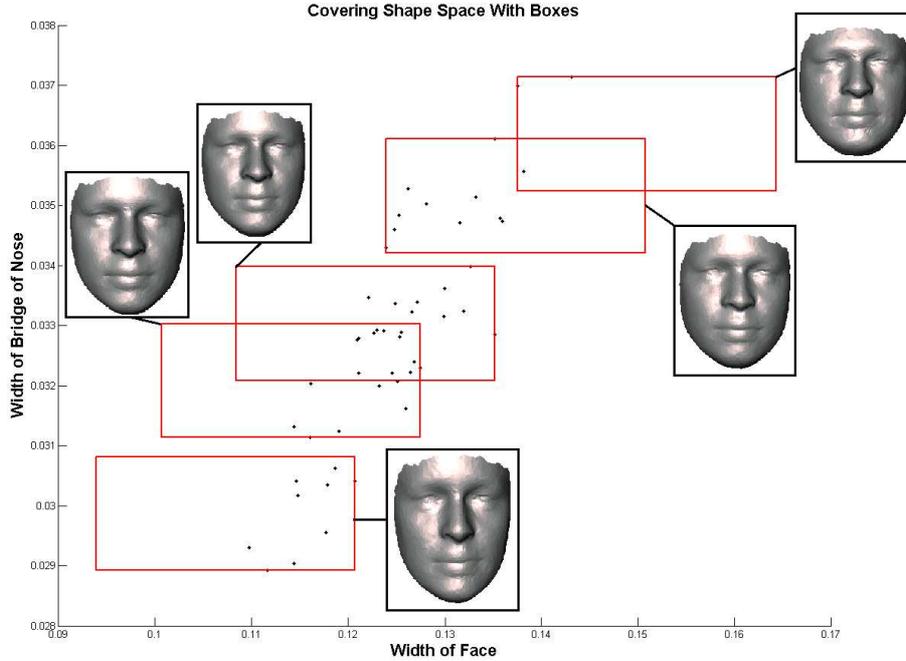


Fig. 3. Covering parameter space with $\epsilon = 1.25$. Unit along axes is meters. Points P are shown as black points, boxes B_i are shown as red boxes, and the design model for each box is shown.

Figure 4 shows the result of a greedy covering of P . Recall that this is a $(4 \log(n))$ -approximation of the optimal solution. The symbols used in the figure are identical to the ones in Figure 3. The left of Figure 4 shows the result when we aim to cover the entire population. We can see that in this example, we only require one extra shape when covering with the greedy algorithm than when covering using a $(1 + \epsilon)$ -approximation with $\epsilon = 1.25$. This shows that the greedy algorithm, which is more efficient and less accurate than the $(1 + \epsilon)$ -approximation algorithm, yields satisfactory results in many cases. The right of Figure 4 shows the result of greedily covering a large subset of P with three boxes. We can see that the selected boxes cover the parameter space well. As before, we can also see that the different computed design models are not merely scaled versions of each other, but differ in their local geometry.

8.2 Design of helmet

When designing a helmet, the three most crucial measurements are the head width, the head depth, and the face height [3]. Using these three measurements yields $\mathcal{S} = \mathbb{R}^3$. Figure 5 shows the measurements on one head in the database. The first dimension measures the Euclidean distance between the red points, the second dimension measures the Euclidean distance between the green points, and the third dimension measures the Euclidean distance between the blue points.

For this example, we conduct an evaluation. We use 1500 heads from the CAESAR database [7] to compute the design models and we then test the quality of fit using 500 different heads from the CAESAR database.

In our example, we aim to design helmets that can be adjusted by 2.99cm in the first dimension, by 3.25cm in the second dimension, and by 3.33cm in the third dimension. This defines the box B .

Figure 6 shows the results of a greedy covering of P . The figure shows the points P as black points and the centers of the boxes B_i as red points. Furthermore, for each box, the figure shows a screen shot of the corresponding Procrustes mean shape. The coordinate axes are shown in the colour of the corresponding dimension (see Figure 5). The left side shows the result when we aim to cover the entire population. This covering requires eight design models. The right side shows the result of greedily covering a large subset of P with three boxes. The boxes corresponding to the three design models cover a

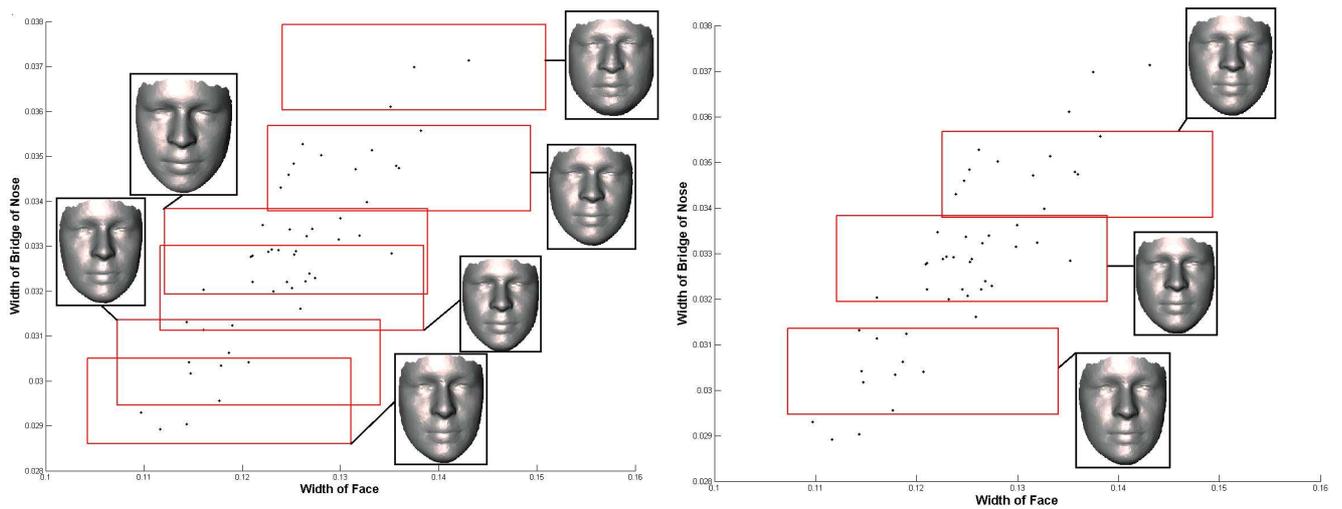


Fig. 4. Greedy covering of parameter space. Left: Complete covering. Right: Covering with three boxes. Unit along axes is meters. Points P are shown as black points, boxes B_i are shown as red boxes, and the design model for each box is shown.

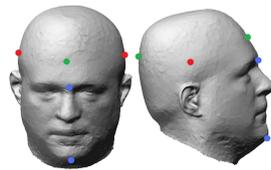


Fig. 5. Measurements used to define parameter space.

large subset of P . Note that for the data used in this example, it is not easy to manually find the best locations of the boxes since we use three measurements and since it is hard to optimally place points manually in three-dimensional space.

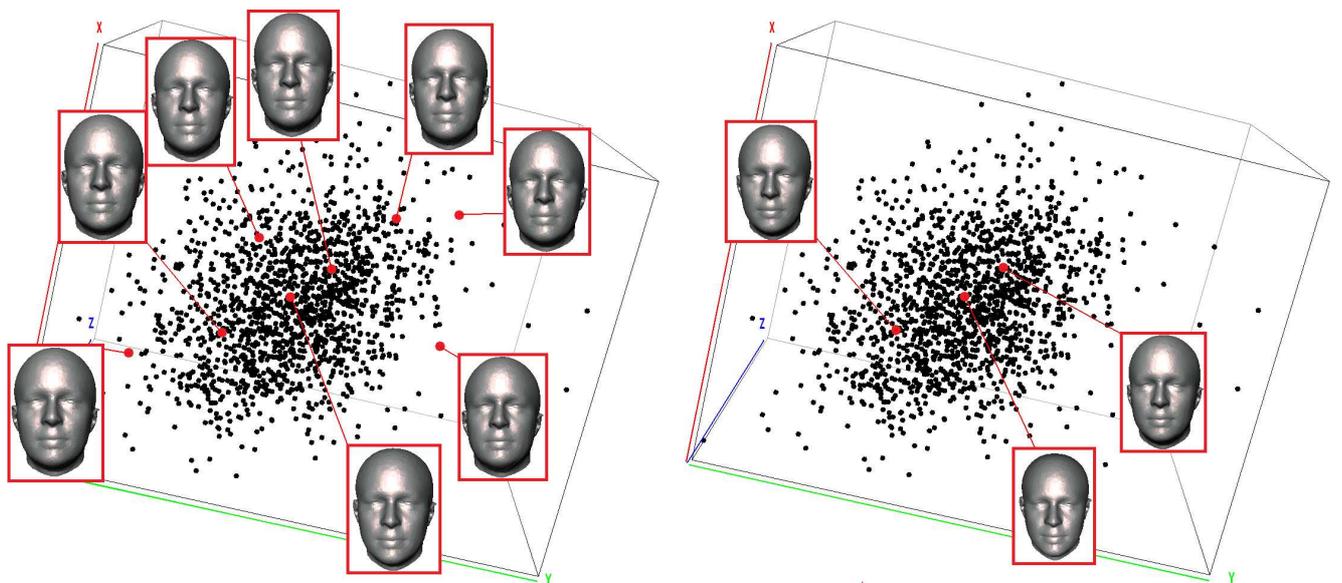


Fig. 6. Left: Greedy complete covering of parameter space. Right: Greedy covering of parameter space with three boxes. Points P are shown as black dots, centers of boxes B_i are shown as red dots, and the design model for each box is shown.

We use 500 different head shapes to compute the quality of fit of the computed design models as follows. We compute the points in \mathcal{S} corresponding to the 500 head shapes and we compute how many of these points are covered by at least one

of the computed boxes. The eight design models computed using the greedy covering algorithm shown in Figure 6 cover 99.8% of all the shapes. The three design models computed by greedily covering the largest subset of P using three boxes shown in the left of Figure 6 cover 95.2% of all the shapes. This shows that when using the three design models shown in the left of Figure 6 to design three sizes of a helmet, we expect that 95.2% of all adults find that at least one of these three helmets fits them.

9 Conclusion

This paper presented a novel approach to generate design models for the design of gear or garments. The approach makes use of the widely available anthropometric databases to find design models that represent a large portion of the population. We find the design models by solving a covering problem in a low-dimensional parameter space.

Note that in this paper, we use translated boxes of the same size to cover the parameter space. This seems the most intuitive shape with which a designer may wish to cover the parameter space. To solve this problem, we presented a $(1 + \epsilon)$ -approximation and a more efficient $(2d \log(n))$ -approximation of the optimal solution. If the aim is to cover the parameter space with translated balls or ellipsoids of the same size, all of the algorithms in this paper can be adapted to this scenario. If the aim is to cover the parameter space with boxes, balls, or ellipsoids of different size or orientation, the algorithms presented in Sections 4.2 and 5 can be adapted to this scenario.

The resulting design models are only as good as the given anthropometric database. This paper discussed the option of improving the coverage of the target population by extrapolating models from the database.

Acknowledgments

We thank Pengcheng Xi for providing us with the data. This work has partially been funded by the Cluster of Excellence *Multimodal Computing and Interaction* within the Excellence Initiative of the German Federal Government.

References

- [1] Kwon, O., Jung, K., You, H., and Kim, H.-E., 2009. "Determination of key dimensions for a glove sizing system by analyzing the relationships between hand dimensions". *Applied Ergonomics*, **40**, pp. 762–766.
- [2] Roebuck, J. A., Kroemer, K. H. E., and Thomson, W. G., 1975. *Engineering Anthropometry Methods*. John Wiley & Sons.
- [3] Meunier, P., Shu, C., and Xi, P., 2009. "Revealing the internal structure of human variability for design purposes". In IEA World Congress on Ergonomics.
- [4] Robinette, K., 2007. Maximizing anthropometric accommodation and protection. Tech. rep., Air Force Research Laboratory.
- [5] Ball, R., 2009. *Size China: An Anthropometric Survey of Chinese Head Shapes*. PhD thesis, Delft University of Technology.
- [6] Dryden, I., and Mardia, K., 2002. *Statistical Shape Analysis*. Wiley.
- [7] Robinette, K., Daanen, H., and Paquet, E., 1999. "The CAESAR project: A 3-D surface anthropometry survey". In 3-D Digital Imaging and Modeling, pp. 180–186.
- [8] McCulloch, C., Paal, B., and Ashdown, S., 1998. "An optimisation approach to apparel sizing". *Journal of the Operational Research Society*, **49**, pp. 492–499.
- [9] Mochimaru, M., and Kouchi, M., 2000. "Statistics for 3d human body forms". In SAE Digital Human Modeling for Design and Engineering International Conference and Exposition.
- [10] Meunier, P., Tack, D., Ricci, A., Bossi, L., and Angel, H., 2000. "Helmet accommodation analysis using 3D laser scanning". *Applied Ergonomics*, **31**(4), pp. 361 – 369.
- [11] Kouchi, M., and Mochimaru, M., 2004. "Analysis of 3D face forms for proper sizing and CAD of spectacle frames". *Ergonomics*, **47**(14), pp. 1499–516.
- [12] Guo, F., Wang, L., and Dong, D., 2009. "Human head 3D dimensions measurement for the design of helmets". In Proceedings of the 2nd International Conference on Digital Human Modeling, pp. 624–631.
- [13] van Kaick, O., Zhang, H., Hamarneh, G., and Cohen-Or, D., 2011. "A survey on shape correspondence". *Computer Graphics Forum*, **30**(6), pp. 1681–1707.
- [14] Ben Azouz, Z., Shu, C., and Mantel, A., 2006. "Automatic locating of anthropometric landmarks on 3D human models". In International Symposium on 3D Data Processing, Visualization and Transmission, pp. 750–757.
- [15] Xi, P., Lee, W.-S., and Shu, C., 2007. "Analysis of segmented human body scans". In Graphics Interface, pp. 19–26.
- [16] Allen, B., Curless, B., and Popović, Z., 2003. "The space of human body shapes: reconstruction and parameterization from range scans". *ACM Transactions on Graphics*, **22**(3), pp. 587–594. Proceedings of SIGGRAPH.
- [17] Wuhler, S., Shu, C., and Xi, P., 2011. "Landmark-free posture invariant human shape correspondence". *The Visual Computer*, **27**(9), pp. 843–852.
- [18] Vazirani, V., 2001. *Approximation Algorithms*. Springer.
- [19] Karp, R., 1972. "Reducibility among combinatorial problems". In *Complexity of Computer Computations*, R. E. Miller and J. W. Thatcher, eds. Plenum Press, pp. 85–103.
- [20] Hochbaum, D., and Maass, W., 1985. "Approximation schemes for covering and packing problems in image processing and VLSI". *Journal of the ACM*, **32**(1), pp. 130–136.