uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Pengcheng Xi

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.C.S.

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A PCA-based Approach to the 3D Reconstruction of Human Body from Single Frontal-view Silhouette

TITRE DE LA THÈSE / TITLE OF THESIS

Wonsook Lee

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

C. Shu

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

P. Bose

R. Laganiere

Gary W. Slater

Le Doyen de la Faculté des études supérieures et postdoctorales / Dean of the Faculty of Graduate and Postdoctoral Studies

A PCA-Based Approach to the 3D Reconstruction of Human Body from Single Frontal-View Silhouette

By

Pengcheng Xi

A thesis submitted to
The Faculty of Graduate and Postdoctoral Studies
in partial fulfilment of
the degree requirements of
Master of Computer Science

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering
University of Ottawa
Ottawa, Ontario, Canada
January 2007

Your file Votre référence
ISBN: 978-0-494-32488-2
Our file Notre référence
ISBN: 978-0-494-32488-2

The undersigned recommend to

the Faculty of Graduate and Postdoctoral Studies

acceptance of the thesis

A PCA-Based Approach to the 3D Reconstruction of Human Body from Single Frontal-View Silhouette

Submitted by **Pengcheng Xi**

in partial fulfilment of the requirements for the degree of

Master of Computer Science

Won-Sook LEE, Thesis Supervisor

Chang Shu, Thesis Co-Supervisor

Chair, School of Information Technology and Engineering

University of Ottawa

2007

Abstract

We introduce a data-driven approach to build a human body model from a single photograph by performing Principal Components Analysis (PCA) on a database of body segments. Our work consists of two main parts, first consistent parameterization of the Civilian American and European Surface Anthropometry Resources (CAESAR) database and second segmented reconstructions of 3D bodies out of 2D photographs.

Analysis on a dataset of 3D scanned surfaces has presented problems because of incompleteness on the surfaces and because of variances in shape, size and pose. In this thesis, a high-resolution generic model is aligned to data in the CAESAR database in order to obtain a consistent parameterization. A Radial Basis Function (RBF) network is built for rough deformation by using landmark information from the generic model, anatomical landmarks provided by CAESAR dataset and virtual landmarks created automatically for geometric deformation. Fine mapping then successfully applies a weighted sum of errors on both surface data and the smoothness of deformation. Compared with conventional methods, our approach has a higher efficiency in making robust alignment. Principal components analyses on segmented bodies display a richer variation than that of the whole body.

Our 3D reconstruction process is composed of training and testing processes. The training aims at finding a relationship between PCA spaces in 3D and 2D. Testing

process then applies this relationship to the 2D PCA-space coordinate of a new image and calculates its coordinate in the 3D PCA-space. Building several PCAs for different body parts, then making separate reconstructions and merging reconstructed body parts create the final results.

We prove from practical experiments that 3D segmented body reconstructions have superiority over whole-body reconstructions. We demonstrate our approach by constructing models of a Caucasian male, an Asian male and a toddler from corresponding photographs based on a Caucasian adult oriented database. Our numerical validation includes 3D reconstructions from silhouettes of Asian males and females, Caucasian males and females, and African males and females, for which we already have their real 3D body information available from the CAESAR database.

Keywords: *deformation, non-rigid registration, CAESAR, RBF, PCA, segmentation*

Acknowledgements

I would like to express my appreciation to my advisors, Won-Sook Lee and Chang Shu. Many thanks for giving me the chance to get involved in the human body modeling. Dr. Lee led me all the way through the project and gave much support and encouragement. Dr. Shu was always there to discuss problems and you provided detailed support and suggestions. Thanks to Ontario Research Network for Electronic Commerce (ORNEC) for providing funding to this project.

My gratitude also goes to the Visual Information Technology (VIT) group at National Research Council of Canada (NRCC) who provided the database of CAESAR and a good working environment. Special thanks to Dr. Zouhour Ben Azouz for your experience in this project. Also thanks Marc Rioux, you are a very nice person to provide the database. Thanks to Jacques Domey and Anita Mainville who helped me to do paper work in VIT. Also thanks Alan Barton for providing your pictures.

Andrew Soon, thanks for providing me your RBF code to get a quick result in the beginning. Also thanks for polishing my thesis. Javier Mora, you did give me good suggestions at group meetings. My thanks also go to other group members.

The most special thanks goes to Lihong, my wife, who was always on my side to give me support and love through the whole process.

Thank you all.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Our motivation is to build a 3D full body model of a person who is unable to maintain an instructed fixed pose for a given amount of time. For example, a toddler is typically non-stationary and unable to follow instructions precisely (when asked to make a specific pose, a toddler can only make an approximate pose). This factor precludes the usage of a 3D scanner (e.g., Cyberware^{TM} body scanners) to accurately capture the subject's full body shape because of the time required to complete the scan (typically at least several seconds). Ideally we would like to be able to acquire the shape data as quickly as possible. Hence, we choose to restrict our input for modeling to be a photograph.

1.1 Statement of the problems

During the past two centuries, anthropometry has been applied to measuring human bodies for analysis of body variations. Traditional measurement instruments include tape (for circumference and length), spreading caliper (for head breadth), anthropometer, sliding caliper, lange skinfold caliper, scale and so on. These instruments were only able to record part of the human body information.

Today, anthropometry is playing an important role in industrial design, clothing and vehicle designs. Distributions of body dimensions are important information in these areas for designing better products for people.

Human space representation poses an increasing requirement for more accurate and complete recording instruments. Take facemask as an example, designers need to know a distribution of the curvatures and surface features for certain population. This cannot be measured with traditional instruments. One possible solution is 3D scanning technology. But processing noisy and incomplete surfaces takes a lot of computation and no general methodologies are available for processing different objects.

Another problem is that scanning results can tell the geometry of disparate subjects but not of a uniform space. So the scanning does not really produce a plausible model. Our assumption is that if we could build a morphable model on a set of data and combine this with statistical approaches, the subjects created by the model would be more interesting.

The difficulty of building a morphable model of this kind is three-fold: first, each dataset has a different number of vertices; second, to be consistent, a model needs to be able to match the features on different data; and third, the model has to fill the holes on the scanned surface that result from grazing angles during scanning.

Reconstruction of 3D data with simple input information has become a hot research topic. Since most 3D scanners are not portable and are always expensive, there is a growing need of acquiring 3D information with simple inputs. When it comes to our body reconstruction, this problem becomes one of reconstructing a 3D human body out of 2D digital images, i.e., the frontal, side or back view images. The pioneering work of Blanz and Vetter [3] introduced data-driven reconstruction of human face, wherein a

single photograph served as the only input required to build a realistic 3D face. Seo et al. [26] adopted the same methodology to accomplish photo-based body cloning.

As one of the most popular statistical analysis methods, Principal Components Analysis (PCA) provides a very efficient way to rank the strength of a dataset's characteristics (e.g., from strongest characteristics to weakest ones); however, a common disadvantage of all data-driven approaches is the difficulty in finding the parameters of an individual who does not belong to the data group. For example, can we reconstruct a child from a database of adults? Can we reconstruct an Asian out of a Caucasian database? Can we reconstruct a new body, which has slightly different pose from those in the database? In addition, the method used by Blanz and Vetter [3] has to solve the high dimensional optimisation problem, which is a bottleneck in terms of computation time and is vulnerable to falling into local minima.

1.2 Solutions and contributions

In this thesis, we propose a two-step alignment framework that combines rough alignment using the RBF network as the first step with fine mapping as the second step. The RBF network helps to align the high-resolution generic model to a target in a very efficient way and removes the markers' error from the second step. Simplified error definition makes easier our control over the selection of weights and parameters. Combining these two steps produces a high degree of efficiency. It also proves to be robust through the alignment of hundreds of targets. We start from the Civilian American and European Surface Anthropometry Resource (CAESAR) project that collected

thousands of range scans of volunteers aged from 18 to 65 in Europe and North America. An analysis on demographics of the CAESAR database is shown in Table 1.

Table 1: Demographics of the CAESAR database (as of a subset of 2391 subjects)

Composition	Percentage	Reported Height (mm)	Percentage	Reported weight (Kg)	Percentage	Age (years)	Percentage
White	76.80%	<1500	1.30%	<40	0.40%	<20	2.60%
African American	11%	1500~1600	11.50%	40~50	3.70%	20~30	22.50%
Chinese	3.50%	1600~1700	29.20%	50~60	17.50%	30~40	25.60%
Filipano	1.20%	1700~1800	33.20%	60~70	21.70%	40~50	26.70%
Japanese	0.70%	1800~1900	19.30%	70~80	21.60%	50~60	16.60%
Vietnamese	0.60%	1900~2000	4.90%	80~90	14.70%	>60	6.00%
Korean	0.40%	>2000	0.60%	90~100	10.20%		
Indian	0.70%			>100	10.20%		
Spanish/Hispanic	2.10%						
Other Mixed Race	1.50%						
Other	1.50%						

For body reconstruction, we propose an approach as shown in Figure 1 for learning a relation and in Figure 2 for reconstruction. The two figures describe the process for whole-body reconstruction, and they also work for the reconstruction of each body part, e.g., the head, the torso, the left leg and right leg, the left arm and the right arm. In Figure 1, one of the two sets of PCAs is on 3D human body meshes (whole-body mesh or body-part mesh) and the other set is on their 2D contours. We then find a relationship D between the two PCA spaces by training on all the selected 3D data, and getting coordinate matrix C_1, and by training on their silhouettes and getting coordinate matrix C_2. In Figure 2, a new body image can easily find its coordinate, c_1^{person}, in the 2D contour PCA space. By usage of the previously acquired relationship D from Figure 1,

we can map this coordinate into the 3D PCA-space and get its coordinate cf^{person}. Then we can reconstruct 3D body mesh with selected 3D eigen-vectors. In terms of reconstructed 3D body parts, we merge them to build a complete and smooth whole body.

There is a contribution in our reconstruction. Previous work on reconstruction of human bodies largely relies on optimisation, which cost much computation; however, our approach put most computation into the training step, and the reconstruction process takes very short time. Moreover, our segmented body reconstruction works very well for subjects in such an age or ethnicity group that it is different from that of the database.

We aim to reconstruct 3D bodies of (i) a Caucasian male, (ii) an Asian male and (iii) a toddler from their front-view photographs correspondingly. Our validation includes reconstructions of Caucasian male and female adults, Asian male and female adults, and African male and female adults.

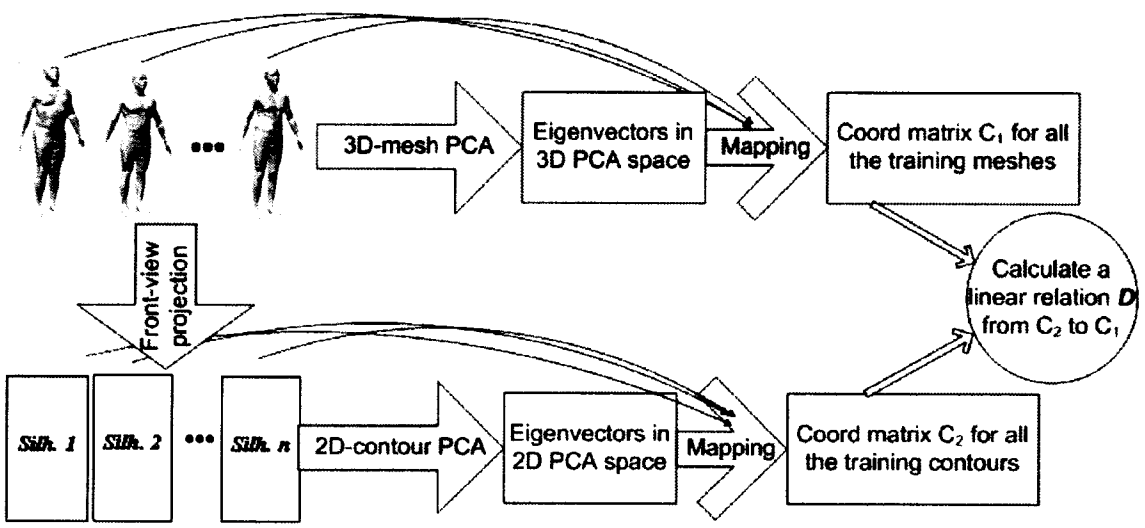


Figure 1: Off-line process for training and getting a relationship from 2D-contour coordinate matrix to 3D-mesh coordinate matrix

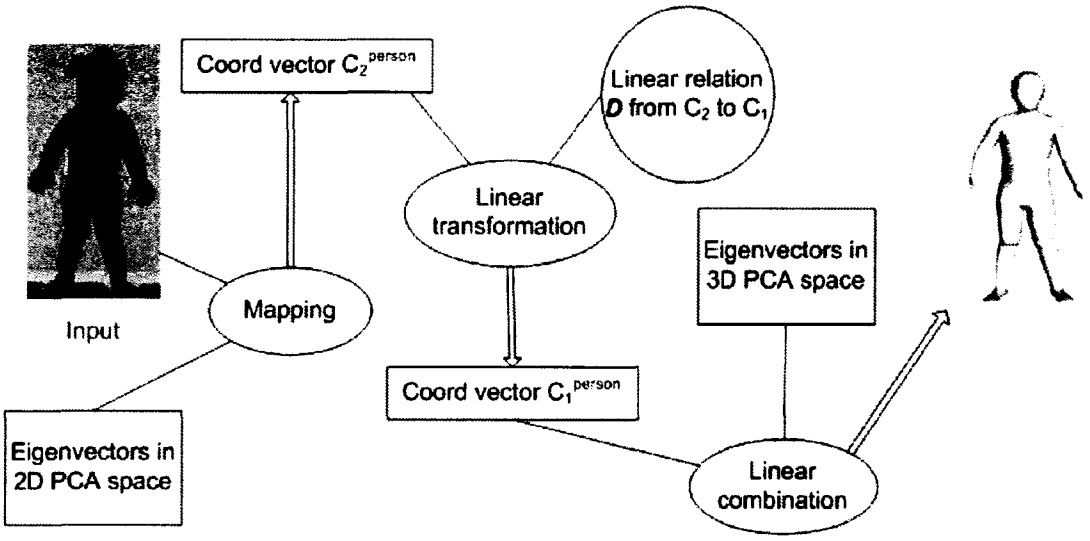


Figure 2: On-line testing process of reconstructing a 3D body out of a 2D contour

1.3 Description of the remaining chapters

The rest of the thesis is organized as follows. A review of literature is conducted in Chapter 2. In Chapter 3, we introduce our two-step approach to parameterization in 3D and a consistent parameterization in 2D. PCA analyses on 3D whole bodies, on 3D segments, and on 2D contours are presented in Chapter 4. In Chapter 5, we learn a relationship from 2D contour PCA space to 3D whole-body and segmented mesh PCA spaces. Then we present our reconstruction results in Chapter 6. Chapter 7 concludes this thesis with discussions.

We would like to make it clear that, when we mention 2D/3D PCA-space in our thesis, it indicates a PCA built for 2D contours/3D meshes.

Chapter 2: Literature Review

In this section, we focus on the literature of modeling body shape variation, finding mutually consistent surface representations (including filling holes in scanned data and registering non-rigid surfaces) and reconstructing 3D objects from 2D images.

2.1 Human modeling and consistent parameterization

A representative work on modeling the variation of human body shape is by DeCarlo et al. [2]. The authors described a system capable of automatically generating distinct, plausible face geometries. They introduced two steps in constructing a face: first, the generation of a random set of measurements that characterize the face, and second, the construction of the best surface that satisfies the geometric constraints imposed by those measurements. The output surface minimizes a measure of fairness, which in this case formalizes how much the surface bends and stretches away from the kind of shape that faces normally have. This is an early work on anthropometrical statistics for likely face measurements, which are later used for constraints on a parameterized surface.

Blanz and Vetter's work in [3] is a milestone for human modeling and reconstruction. It explores two key problems of computer-aided face modeling. First, new face images or new 3D face models can be registered automatically by making a large number of comparisons with an internal model. Second, the approach regulates the naturalness of

the faces modeled and avoids those with an "unlikely" appearance. This is the work that successfully includes statistical approaches into computer graphics as a way of obtaining visual information from a given database (as Figure 3 shows). Blanz and Vetter aligned features using a modified version of 2D optical flow because their head scans have a cylindrical parameterization. The system also enables users to create new characters and to modify facial attributes by varying the model's coefficients. Within the constraints imposed by prior probability, the range of possible faces varies greatly, and all linear combinations look natural.

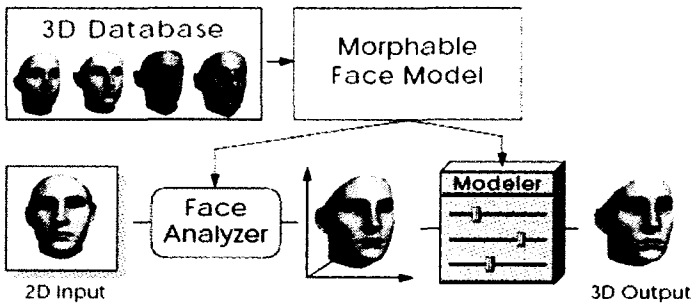


Figure 3: Building a morphable model and reconstructing new 3D models with 2D input [3]

Whole-body parameterizations are more complicated because the human body does not easily lend itself to cylindrical mapping [4]. Praun et al. [5] therefore introduced an approach that builds an n-way correspondence between meshes of the same topological type by the use of feature markers. This gives immediate correspondences between models and allows remeshes; however, this approach only applies to complete surfaces. In datasets like ours, where some holes exist, the parameterization will not work well. Also, since this parameterization relies greatly on landmarks, problems arise when markers are missing. Lee [28] established consistent parameterization of human body

shape from different kinds of inputs, e.g., a scanned data, an artistic model or a set of photographs, while the system needed a specially structured generic model.

Some work on filling holes in the scanned surface has been done, including [6]. The authors' solution starts by constructing a signed distance function, the zero set of which is defined only in the vicinity of observed surfaces. A diffusion process then extends this function through the volume until its zero set bridges any holes that may be present. Other hole-free reconstruction methods, such as [7] and [8], have the desirable feature of filling holes smoothly. This is good for areas like the top of the head; but for the soles of the feet, the approach fails to make them stable, and instead makes them shrink to the upper surface. In addition, not all the approaches are able to fill the holes between two legs: filling holes easily bridges the legs.

In [9], Allen and his colleagues started from a subdivision template that resembles the range surface. They re-parameterized the surface by sampling it along the template normals to construct a set of displacement maps (as Figure 4 shows). Then they were able to fill in displacements smoothly. The soles of the feet were therefore filled in flat. A related displacement-mapped approach was also developed in [10]. But a pre-condition for avoiding the crossing of sample rays is that the template surface is already a fairly close match to the original. The latest work on filling holes is from Kraevoy and Sheffer [36]. They introduced a mesh completion approach using a mapping between the incomplete mesh and a template model. The mapping is computed by a bijective parameterization of meshes with gaps and holes. The completed models have the same topology as the template.

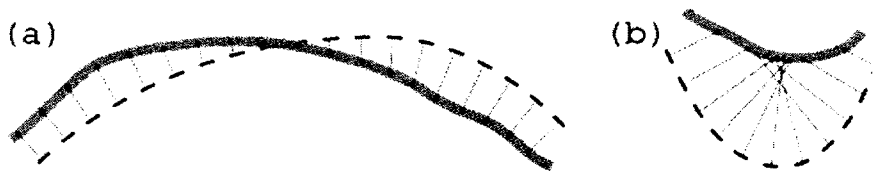


Figure 4: (a) To construct a displaced subdivision surface, Allen et al. cast rays (red arrows) perpendicular to the template subdivision surface (dashed blue line) to the nearest scanned surface (thick gray line). Because the direction of each ray is determined by the subdivision surface, they need only record the distance. (b) If the template surface is too curved and the scanned surface is too far away, then the rays can cross, causing the parameterization to fold over on itself. This can be avoided by ensuring that the template surface is close to t he scanned surface. [9]

The following work applies generic models to alignment as well as to filling holes. One of the latest contributions comes from Kahler et al. on the "Head Shop" [11], which presents a versatile construction and deformation method for head models with anatomical structure. The paper employs this technique to fit a generic head model to imperfect scan data, and to simulate head growth from early childhood to adult age (as Figure 5 shows). Their deformation uses volumetric radial basis functions.

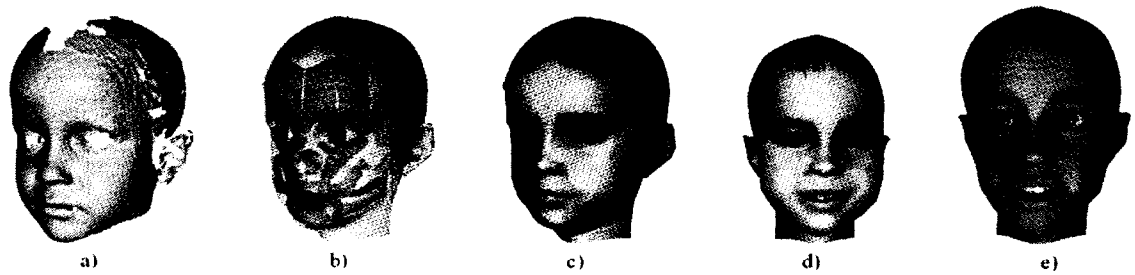


Figure 5: Head models generated from a range scan of a five-year old boy: a) scan data; b) adapted, animatable head structure; c) textured; d) age changed to one year, smiling expression; e) age changed to 20 years, surprised expression. [11]

Anguelov et al. [12] presented a framework for determining complex shape models from range scan data. Their framework consists of algorithms based on the theory of

probabilistic graphical models. This allows complex shape models of different objects and object classes to be learned with minimal human intervention.

Another RBF–based approach is from [13], where dense surface correspondences are computed by volume morphing with RBF followed by a cylindrical projection. Volume morphing roughly aligns features of the two models such as eye sockets, nose ridge, lip corners, and chin points.

Seo and Magnenat-Thalmann [14] designed a system (as Figure 6 shows) that is close to Thomas Vetter's face system [3]. Starting from 3D data, they derived by RBF the deformation functions that generate the appropriate shape and proportion of body geometry [14]. They applied a skeleton-driven approach to the rigid deformation of a generic model to target body surfaces, and used collision detection for fine skin fitting. The skeleton-driven approach is from skin deformation [15]. Vertex blending and skeleton deformation create smooth deformations. Simple vertex blending works very well for small angles up to perhaps 60 to 90 degrees, but can fail badly at angles closer to 180 degrees.

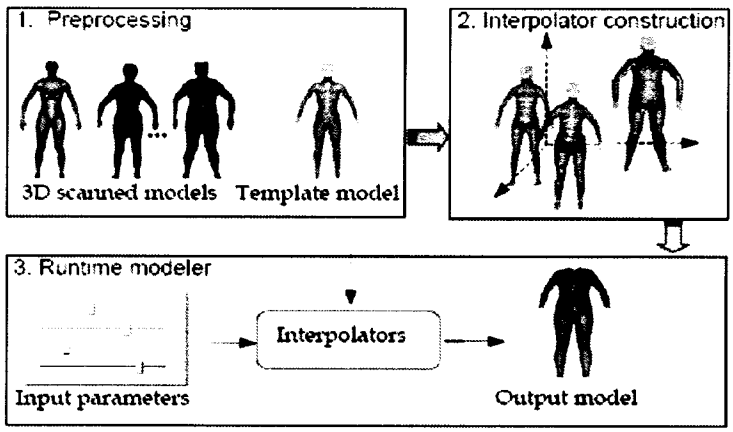


Figure 6: An overview of body modeling from sizing parameters [14]

Hitherto, the generic model (or template model) approach has been commonly used for human modeling. The matching technique used in [4] is based on an energy-minimization framework, and is similar to that of Marschner et al. [17]. Instead of using the surface smoothness term in [17], the authors in [4] tried to minimize variation of the deformation itself, so that the holes in the mesh are filled in with detail from the template surface. This keeps local warping stable so that the whole surface gets little deformation in local areas.

In an earlier work, Feldmar and Ayache [18] described a registration technique with matching surface points, normals, and curvature that maintain a similar affine transformation within spherical regions of space. The authors in [4] have the same idea by using "locally affine deformations". The difference is that they do not use surface normals or curvature, but directly define the smoothness term over the surface rather than over a spherical volume [18].

In our research, we borrowed the idea of RBF and the definition of weighted errors, from previous work, and tried to combine them to achieve better alignment results in a template-based method. Our RBF-based error minimisation method is twice as fast as the method by Allen et al. [4].

2.2 Reconstruction out of simple inputs

After building a morphable model with a consistent parameterization of 3D data, we discuss current approaches applied to reconstructing 3D humans out of 2D images.

From the work of Blanz and Vetter [3][25] to the work of Seo, Yeo and Wohn [26], most approaches are based on minimising a defined difference between the morphable

3D model and the target images. Blanz's work discussed a high-dimensional minimisation problem, which considered model parameters and rendering parameters. The latter one contains camera position, object scale, image plane rotation and translation, intensity of ambient light, and intensity of directed light.

Pighin et al. [24] achieved high-realism for the synthesis of faces and facial expressions from photographs (as Figure 7 shows). Model fitting was again used to adapt a generic model to fit an individual's face and facial expression. Using orthogonal photographs as input, Lee et al. [16][27] successfully applied geometric-based deformation to face and body modeling and animation.

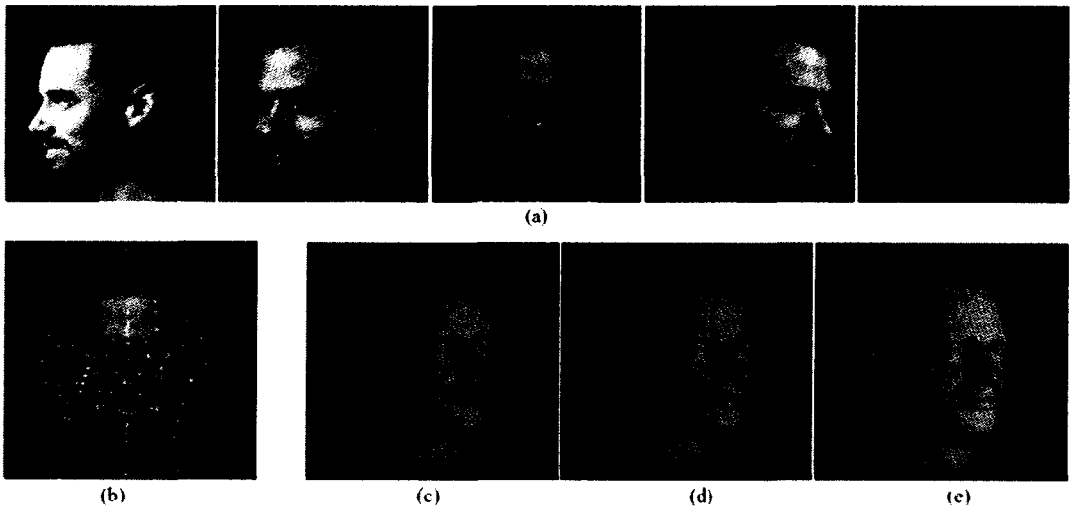


Figure 7: Model fitting process: (a) a set of input images with marked feature points, (b) facial features annotated using a set of curves, (c) generic face geometry (shaded surface rendering), (d) face adapted to initial 13 feature points (after pose estimation) (e) face after 99 additional correspondences have been given. [24]

A latest work from Seo et al. transferred this optimisation scheme to reconstruction of human bodies [26]. This method differs from previous ones in that it did not consider the texture during model fitting. The texture image was later mapped to the reconstructed 3D

model using feature correspondences (as Figure 8 shows). Therefore, their optimisation is based on the minimisation of the difference between 3D and 2D silhouettes.

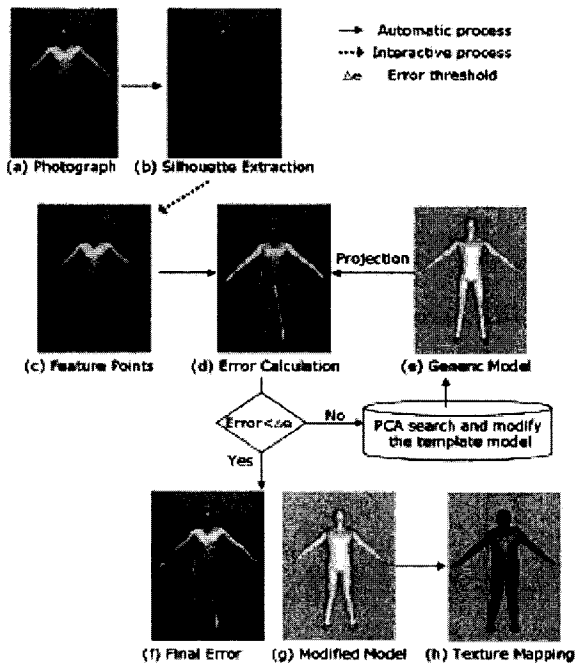


Figure 8: Overview of a 3D body reconstruction from photos based on range scan [26]

Approaches based on optimisation have three disadvantages: low efficiency (a minimisation problem of high dimensions), possibly local minimisation, and need of manual initialisation. In terms of the difficulty of parameter manipulation, we also talk about a learning method based on survey results for reconstructing 2D faces in [29].

We find from [1] that some relations can be learnt from training database. In this earlier work from Vetter and Poggio, they tried to generate new views of a 3D object from a single real image. Linear transformations from one view image to another view image was learnt and justified from a basis set of 2D prototypical views.

In respect of the disadvantages from the optimisation approach, we thus propose a learning scheme for the mapped weights of the 3D and 2D PCA-spaces. We are assuming

a linear relationship between the mapped weight vector along 3D eigen-vectors (coordinate in 3D PCA-space) and along 2D eigen-vectors (coordinate in 2D PCA-space). For a new image, the coordinate in 2D PCA-space can be calculated after mapping its contour onto the 2D contour PCA-space. Using the previously calculated linear relation, this coordinate is easily mapped to 3D PCA-space. A linear combination of selected 3D eigen-vectors using this mapped coordinate creates a well-reconstructed 3D model. Our approach in 2D is based on the contours only.

We make another contribution by segmenting the whole-body database into body parts. Our methodology has a richer expressiveness than conventional PCA approaches because it divides the body into several parts (head, torso, two arms and two legs) and builds a separate PCA for each body part. It aims to model an Asian and a child from the Caucasian adult database. Component-based approach has recently become popular in face recognition research. The premise behind this approach is basically to divide a face into several components and build separate statistical classifiers such as support vector machine, PCA and linear discriminant analysis. Several papers [30][31][32][33] proved that component-based analysis raised the face detection and recognition rate over conventional whole-face recognition, especially on expressed faces. Face shape usually changes as a whole structure (e.g., a facial expression moves whole regions of the face) while body shape changes as separate structures. For instance, the muscle tone and amount of fat in areas such as the legs, the arms and the abdomen vary depending on daily behaviours, so it is sensible to treat the face as a whole part and the body as several individual parts when we do 3D mesh modelling.

Chapter 3: Consistent Parameterization in 3D and 2D

To conduct a statistical analysis on the CAESAR database, a consistent parameterization for all the data must be done in advance. This consistent representation for each 3D mesh ensures that every datum has same number of vertices and each vertex represents the same feature in different bodies. To achieve this goal, a generic model is introduced to align each datum by a two-step scheme: an RBF-based rough alignment plus a fine alignment with a minimization of errors.

The initiative to find the relation between 2D and 3D imposes another requirements on representing 2D contours in a consistent way. To conduct PCA on 2D contours, we need first project 3D data to 2D, get their silhouettes and contours, and then try to make a consistent representation by mapping 3D landmarks to 2D space. For representing contours for testing images, we need manually put 2D landmarks according to textures.

3.1 Consistent parameterization for 3D meshes

3.1.1 RBF-based rough deformation

RBF network builds upon two sets of landmarks, one is on the generic model and the other is on target mesh. Other vertices on the generic model will be mapped with interpolation in terms of their distances to the landmarks on the generic model.

Landmark definition

The deformation function for the RBF network comes from Bookstein [19]. He advocated an approach based on thin-plate splines. The RBF network is entirely built on landmarks. With a number of landmarks in both the generic model and target surfaces, good interpolation is ensured. From Figure 9 to Figure 11, we identify 73 anatomical landmarks on our generic model that correspond to those on target surfaces palpated through the skin. Experts placed these landmarks near skeleton joints; in CAESAR datasets, each person was scanned, giving thousands of data items containing this information. Our approach starts with a high-resolution model with 328K triangles.

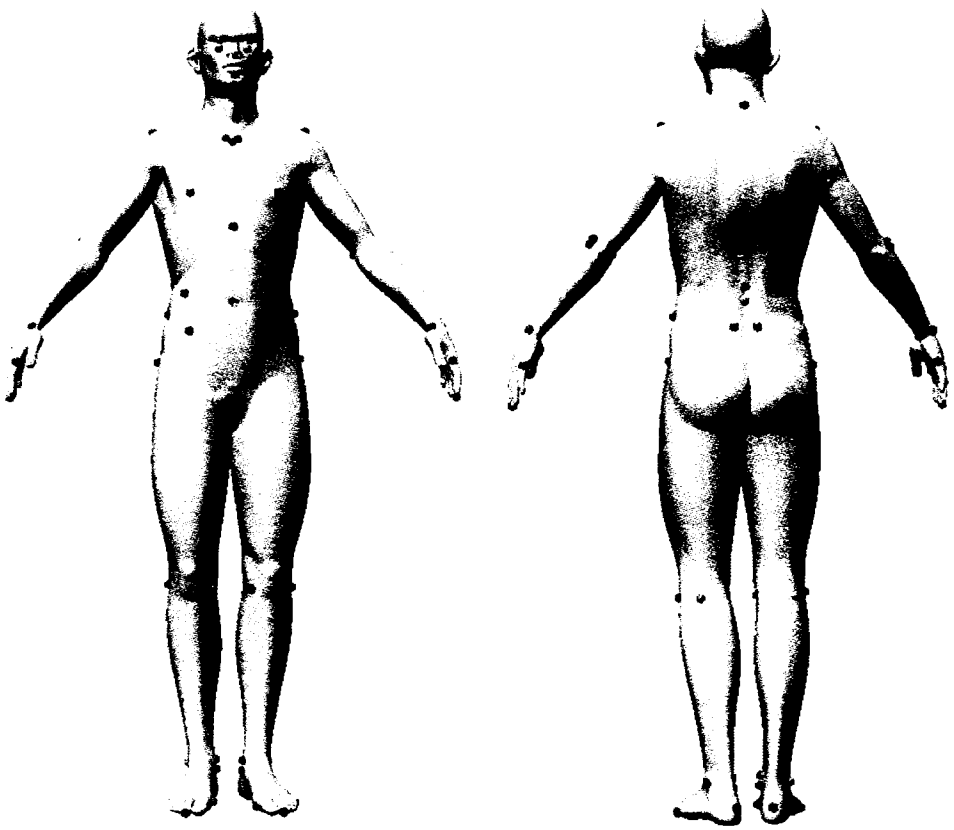


Figure 9: Locations of the 73 landmarks on the 328K generic model

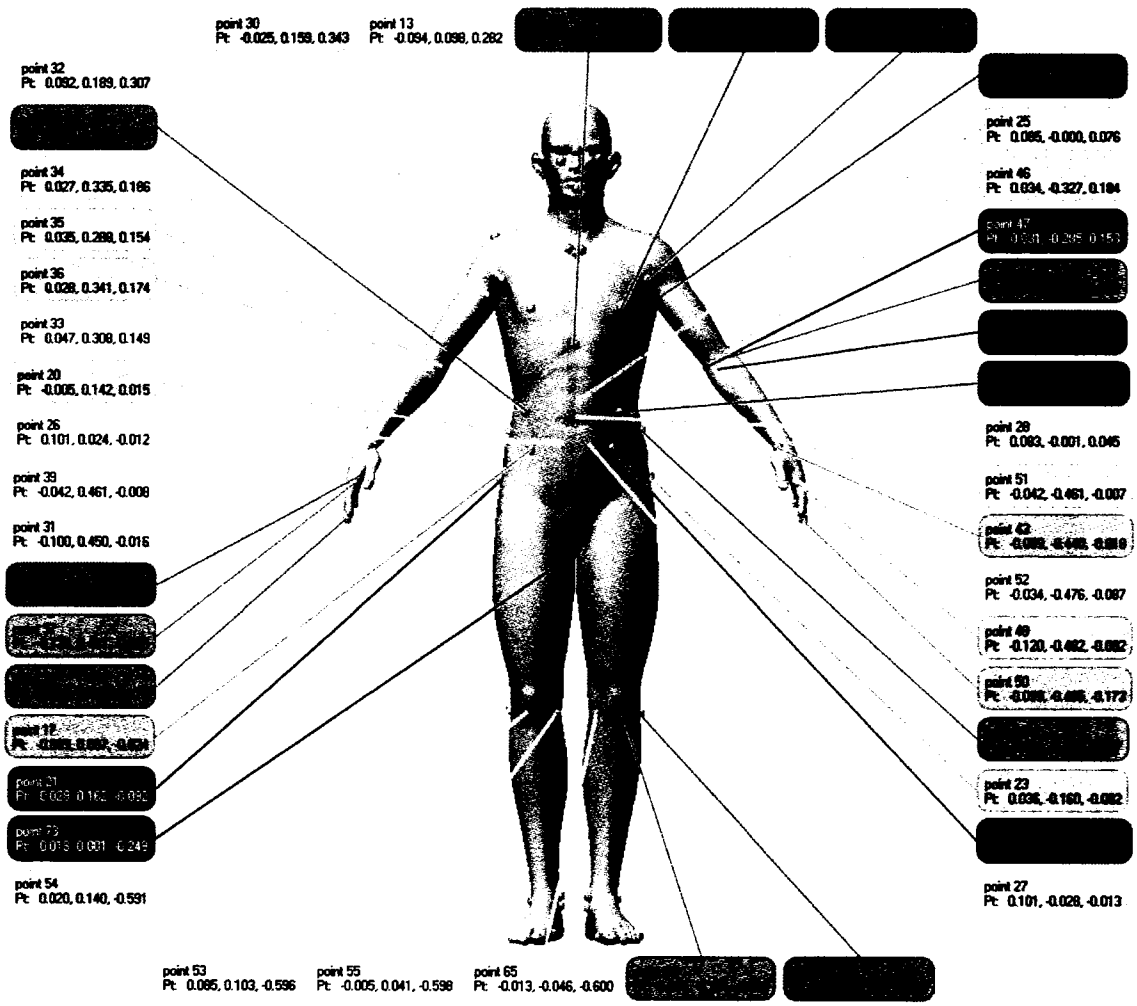


Figure 10: Landmark names and coordinates on frontal view of the 328K generic model

Besides these 73 landmarks, we define another 12 geometrical landmarks on both the generic model and each target surface so that the model becomes closer to the target surfaces after RBF-deformation. The added landmarks are very close to the surface but not necessarily on the surface. They were added to maintain the volume of specific body parts for RBF-deformation. We do so on the upper arms, the upper legs and the lower legs where current landmark information shows that RBF has little influence on

deformation. The virtual landmarks are obtained from anatomical landmarks using linear interpolation or extrapolation.

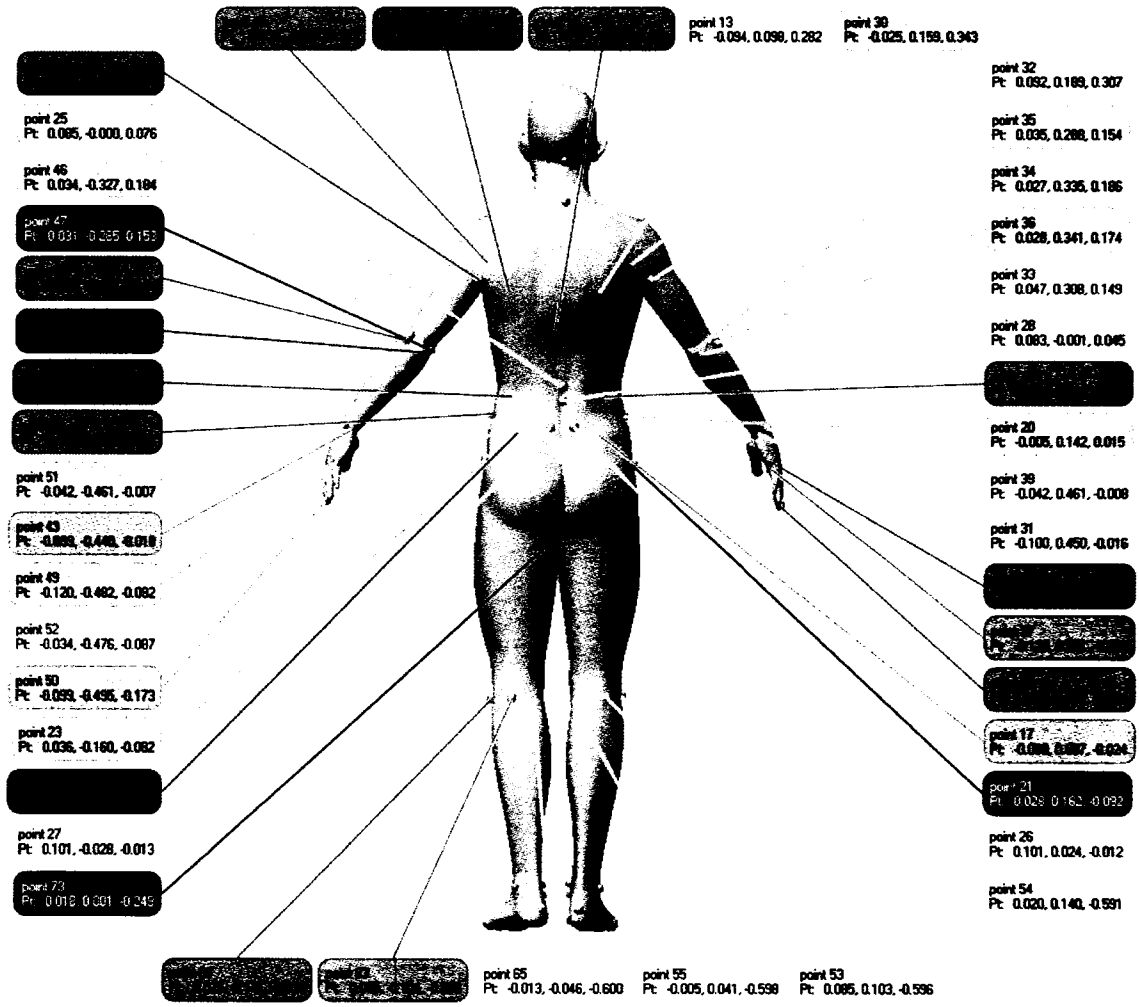


Figure 11: Landmark names and coordinates on back view of the 328K generic model

Radial Basis Function network

The problem of building a RBF network can be stated as a problem of interpolation: let $p_i \in \Re^3$ and $q_i \in \Re^3$, $i=1...n$, be two sets of n landmarks (in our case, $n=85$), which serve as the input. The source landmarks p_i lie on the generic model and target landmarks

q_i correspond to features on the target surface. Three RBF networks on x, y and z coordinate respectively, are established to build the mapping:

$$q_i = f(p_i),\ i = 1,...,n, \tag{1}$$

Since this function is defined over the volume spanned by the landmarks, it can be used to deform all vertices on the body surface. This mapping can then be expressed by a radial basis function, i.e. a weighted linear combination of n basic functions defined by the two sets of landmarks:

$$f(p_i) = \sum_{j=1}^{n} w_j \Phi_j(p_i) \tag{2}$$

When computing the mapping coefficients, input points will be the landmarks and when doing the deformation, input points will be the vertices in influence region of the landmarks from the high-resolution generic model with 328K triangles.

This linear system is solved using a standard LU decomposition with pivoting. After training and computing the weight vector, new positions of those non-feature vertices are calculated by using this RBF network with their initial positions. The RBF transformed model maintains the same topology as the generic model.

Comparison with different radial basis functions

In the mapping defined by Equation (1), there are several radial basis functions, which could be applied for model deformation. We first introduce a thin-plate spline function:

$$\Phi_j(r) = r^2 * \log(r), \tag{3}$$

where r is the Euclidian distance between the feature point and the input point [20].

Figure 12 presents one of the targets the generic model is deformed to.

The second radial basis function introduced is a multi-quadrics function:

$$\Phi_j(r) = \sqrt{r^2 + s^2} \tag{4}$$

where s is called a stiffness constant, which controls the local or global effects of the landmarks. By increasing the value of s from 0 to 1.0, the RBF-transformed generic models are presented in Figure 13. After this increment, the face has a better deformation while the legs start to get twisted. We found from further experiments that, if the nose sticks to the face in the rough-alignment process, it would not be able to recover its shape after fine alignment. Therefore, if we select the multi-quadrics function for RBF, we need to select an optimal value of s so that the face shape could be well maintained and the lower legs could get less twisted.

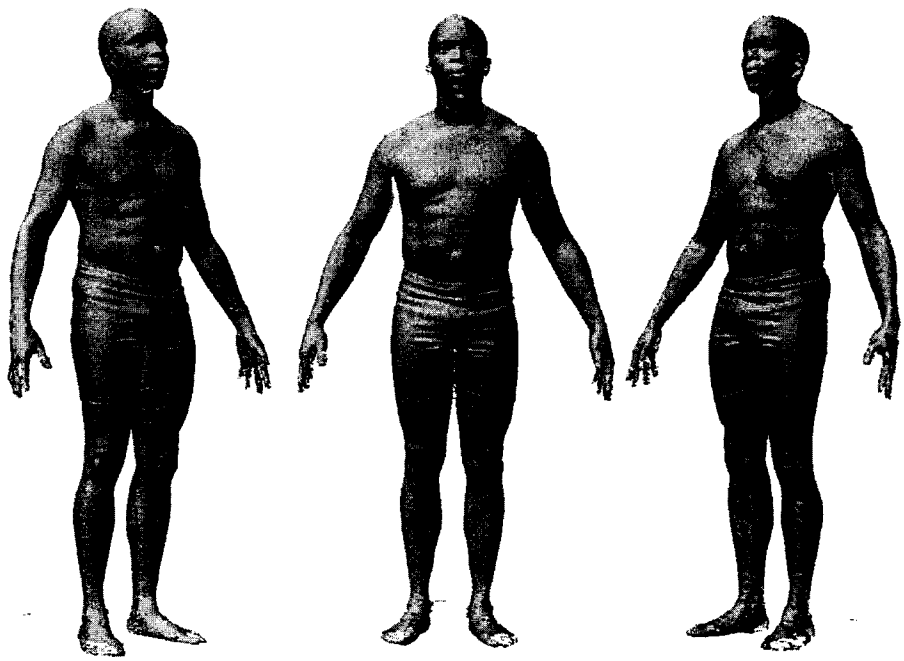


Figure 12: Snapshots of one of the targets ("csr0232a" in the CAESAR database)

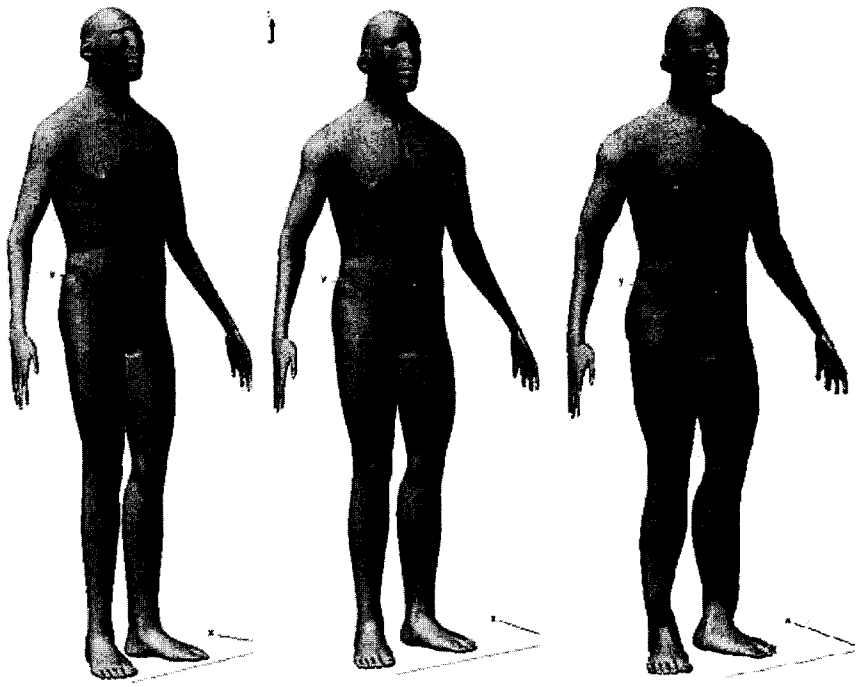


Figure 13: Deformed generic models with a multi-quadrics radial function (from left to right, the value of *s* is selected to be 0, 0.005 and 0.01)

We introduce the third radial basis function, a Gaussian function:

$$\Phi_j(r) = \exp(-r^2 / c^2). \tag{5}$$

It has the only parameter c. Experimental results from selecting different values of c are demonstrated in Figure 14. It is found that while increasing the value of c, deformation on pose is attenuated and the surface becomes coarser. This tendency could be explained from the definition of radial basis function. With the increment of c value, the value of radial basis function increases. Accordingly, there is a long "Gaussian" distance between current vertex and the influencing landmarks, which means the deformation from these landmarks will be attenuated.

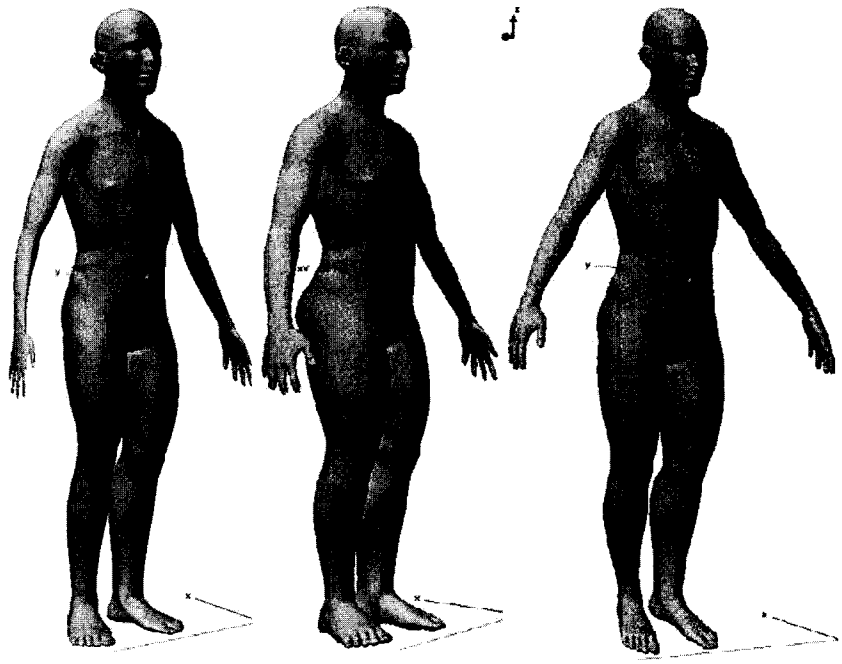


Figure 14: Deformed generic models with a Gaussian radial function (from left to right, the value of *c* selected is 1.0, 10.0 and 100.0)

By comparison of the deformed generic models with three radial basis functions in Figure 15, the Gaussian function with the parameter value of "1.0" was actually selected for our project. One main reason is that Gaussian function has the best performance among three functions in maintaining human-alike shape after the rough deformation.

After RBF transformation, the generic model has been deformed closer to the target model in shape, pose, and height (see comparisons in Figure 16 for difference before and after RBF-deformation). RBF ensures a good registration of landmarks in that all landmarks have been mapped to corresponding targeting positions. Therefore, in second-step fine alignment, we discard marker information and concentrate only on data and smoothness errors.

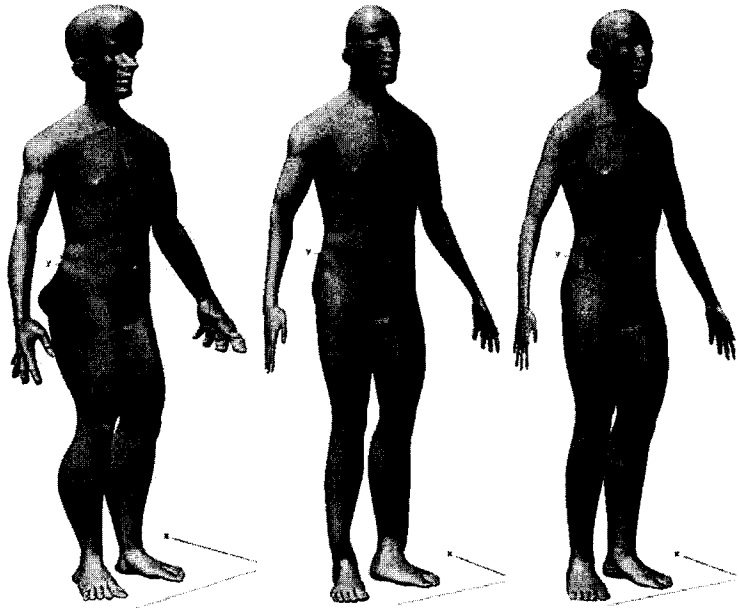


Figure 15: Deformations with different radial basis functions (from left to right: thin-plate, multi-quadrics and Gaussian)

RBF has several practical advantages. First, if the original generic model is significantly different from the target, the RBF can provide translation, rotation and scaling for the rough alignment in an efficient way. Second, the normal information of most vertices can be discarded because the transformed surface is already close to the target. Experimental results will be shown to prove that RBF moves the generic model faster than the marker error did in [4].

Figure 16: Comparisons between the target "csr0232a" and the generic model (left: original generic model in grey and the target, right: RBF-deformed generic model and the target)

3.1.2 Fine alignment

After the rough transformation, we only need concentrate on a greatly reduced difference between the transformed generic model and the target. It must be emphasized that a still high difference between the two models will result in a local minimum in the fine mapping process. By discarding markers' errors, the error definitions in [4] are simplified

to two: (a) the difference between the source surface and target surface (data error) and (b) the difference between transformations on neighbouring vertices (smoothness error).

Suppose the transformed generic model is U, and the scanned target surface is Γ. For each vertex $v_i (i = 1...N)$ on U, we define a 4×4 transformation matrix T_i. Each vertex has 12 degrees of freedom to define the transformation. Our intent is to find a set of transformations that move each vertex to its closest point on Γ. This process needs to be repeated several times in order to align the RBF-transformed generic model to the target.

3.1.3 Error minimisation scheme

The evaluation of these repetitions depends on two errors: data error and smoothness error. The data error is defined as:

$$E_d = \sum_{i=1}^{M} dist^2(T_i v_i, \Gamma), \tag{6}$$

where $M(<N)$ is the number of non-marker vertices in U, and *dist()* function computes the Euclidean distance between a vertex on U and its closest vertex on Γ.

Since the generic model has 328K triangles, we use an "Approximate Nearest Neighbour" searching library [22] to identify the closest vertices in an efficient way.

Smoothness error is defined as:

$$E_s = \sum_{\{i,j\}\{v_i,v_j\}\in edges(Y)\}} \| T_i - T_j \|^2 \tag{7}$$

The smoothness error is not defined for the smooth surface but for the actual deformation applied to the generic model. The diffusion process of transformations

extends this deformation to neighbouring vertices. This results in that deformations for an area of nearby vertices are maintained as similar as possible.

The definition of smoothness error minimizes the deformation over the template surface and thus prevents adjacent parts of the template surface from becoming aligned to disparate parts of the target surface.

The overall error is defined as a weighted sum of these two errors:

$$E = aE_d + bE_s, \tag{8}$$

where a and b are two weights for the data and smoothness errors.

By setting two different weights for data and smoothness errors, and by minimizing the overall error in (8) after running the optimization using L-BFGS-B [23], we get the fine mapping results in Figure 17. From three different viewing angles, the fine-aligned generic model looks the same as the target shown in Figure 12.

In programming, some special treatments need to be taken into consideration. To avoid the two legs get bridged, normal information of each vertex should be calculated for finding a compatible targeting vertex. When aligning the generic surface to the target where data is missing, we get an evenly distributed result by keeping vertices static if their nearest neighbours lie at the boundary of holes. Also during the fine mapping step, both of the hands and ears are specially treated by finding their index in the whole model and assigning a lower weight to calculation of the data error for vertices in these meshes.

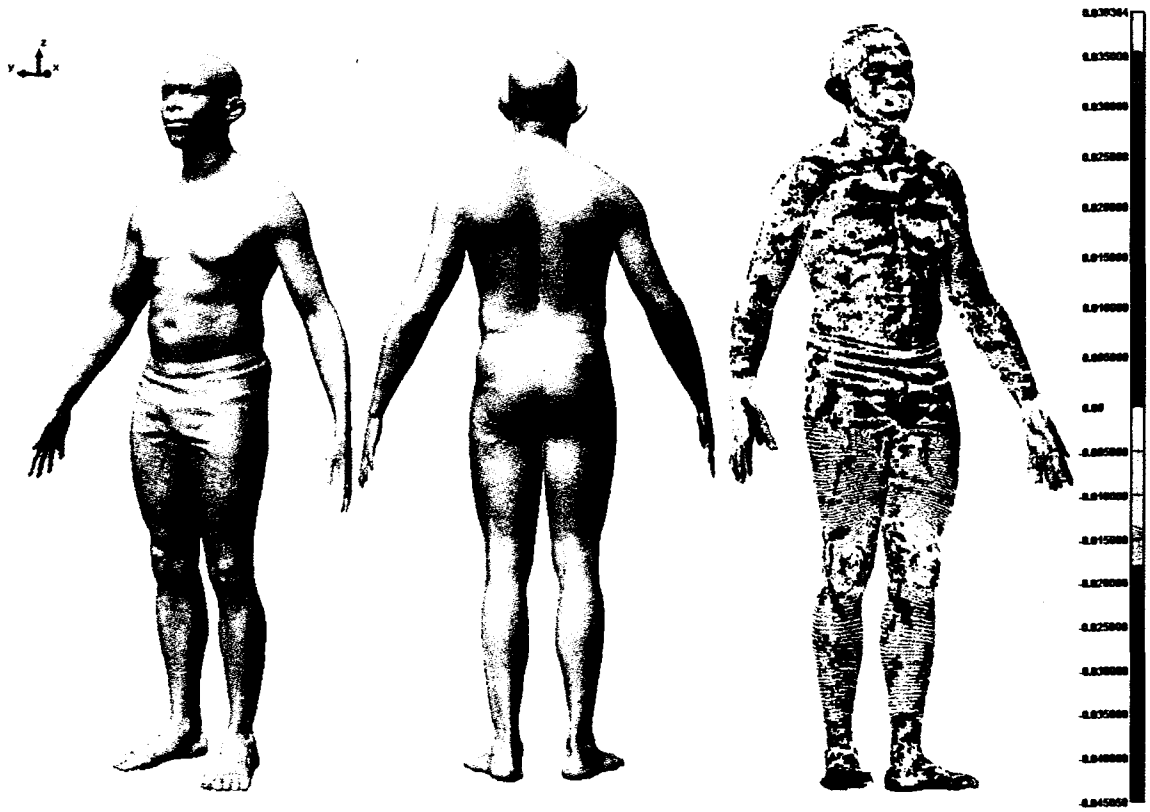


Figure 17: Fine-mapping results for "csr0232a" (from left to right: diagonal view, back view, and a comparison between fine mapping result and the target)

3.1.4 Analysis of results and declaration of contributions

With different sets of weights, Figure 18 shows a comparison of results. The fine mapping result on the right side has artefacts (scratch mark) on the surface. Figure 19 shows the detail: similar transformations on the irregular topology of the generic model create the ridge on the fine mapped result (regular topologies on the left side and on the right side of an obvious vertical division line in the generic model create an irregular topology. When two deformations from both sides meet at the line, a ridge is created). By setting a higher weight for the data error, this artefact is removed.

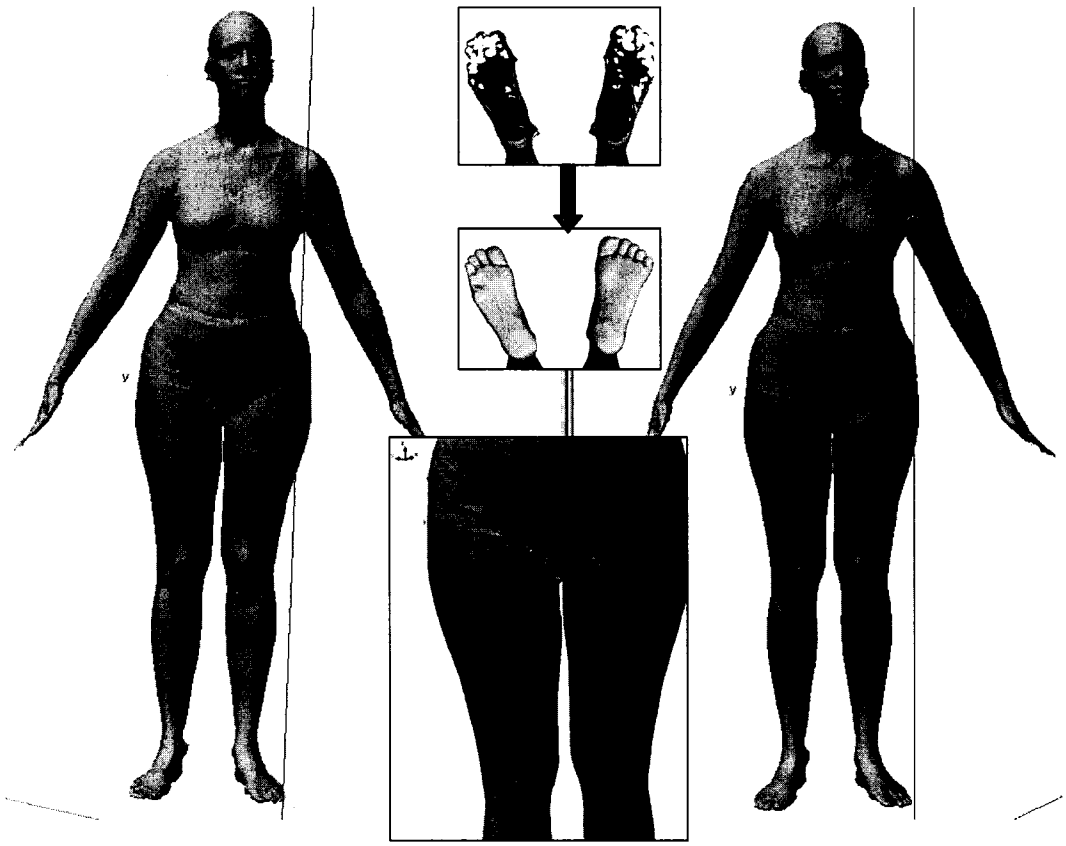


Figure 18: Fine mapping results from two sets of weights: higher data weight (left: a=0.96, b=0.04) vs. higher smoothness weight (right: a=0.71, b=0.29); in the middle: feet soles repaired and smoothed, two legs get de-bridged.

We also implement part of the approach used by Allen et al. [4] to find the deformation ability of the defined smoothness error. Taking into account the marker and smoothness errors only, we show a result as the middle image in Figure 20. A close look at the image patch shows that, with a high-resolution model, the markers' error plus the defined smoothness error have local deformation ability only. Vertices being far from markers will not get deformed as expected. In Figure 20, since the marker at the crotch between two legs does not help deform neighbouring vertices, a fine-mapped result by adding the data error displays bridged legs.

Figure 20 also explains the reason why Allen et al. used a multi-resolution approach [4]. There will be a better diffusion performance from the defined errors if the generic model is first deformed in a lower-resolution. However, this will create another problem in deciding the resolution: a simplified generic model with a still high resolution will have the same problem in disseminating the deformation from markers; in a generic model with a too low resolution, deformations of certain vertices affected by nearby markers might get mixed up and the result will becoming worse after more iterations. Also for a different generic model, there is no general rule that can help decide a proper resolution for a lower-resolution generic model.

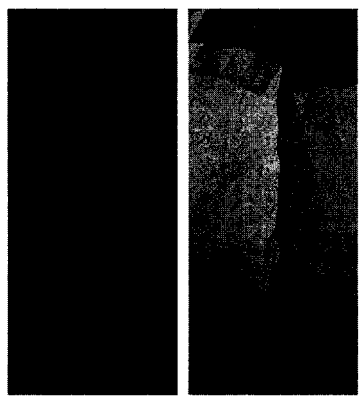


Figure 19: Irregular topology on generic model (left) creates artefacts on the final mapping result (right)

For comparison, we also fully implement the multi-resolution method from Allen et al. [4] to the same target (about 1.71m in height as recorded in the database). Table 2 compares the two approaches, and shows that, after convergence, the RBF-based fine matching gives more accurate results than the two-resolution approach in [4]. Both of the experiments are tested on a PC with 3.1GHz frequency and 1GB memory.

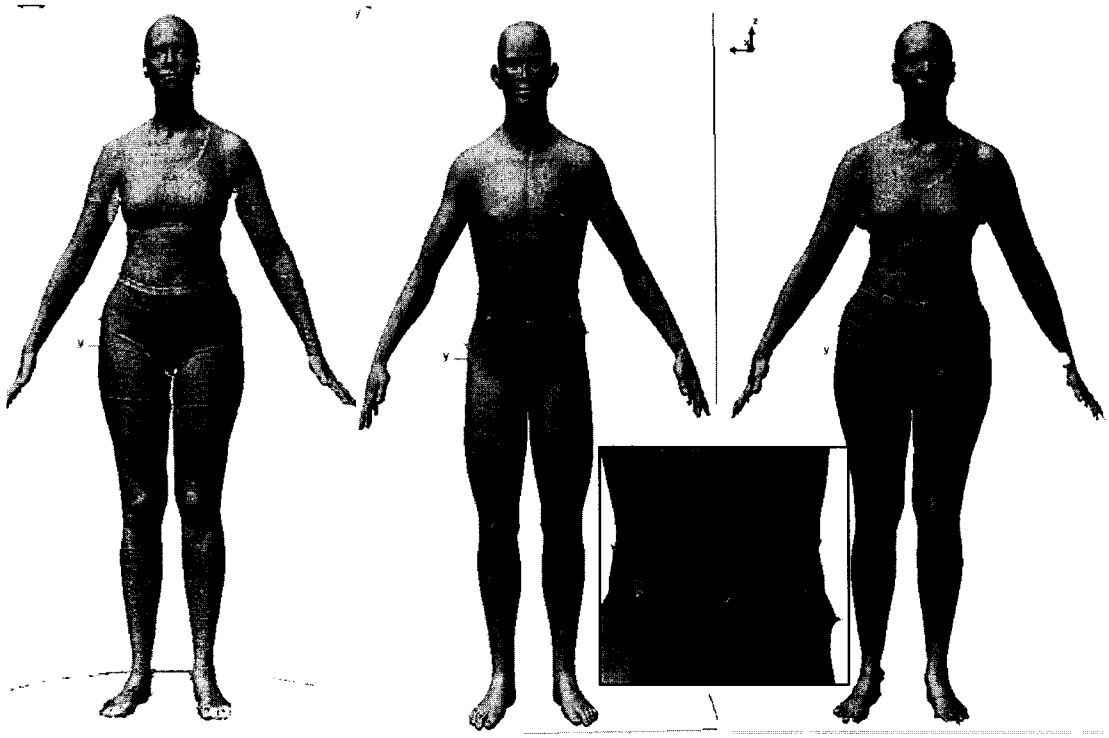


Figure 20: Deformation results (from left to right: the target, alignment result with smoothness and markers' error, fine mapping result)

Table 2: Summary of the comparison on final errors with two approaches when the final errors in both methods make almost no changes (for our approach: 200 iterations are needed; method in [4] needs at least 800 iterations)

	RBF plus fine mapping (ours)	Multi-resolution [4]
Number of points	143522	143522
Mean error (m)	-0.000153	0.000154
Standard deviation (m)	0.005998	0.006129
Root mean squared error (m)	0.006000	0.006131

By setting a threshold of the root mean squared error to be 0.006200 m, we measure the time and iterations needed for each approach. Table 3 shows a great advance over the method in [4]. Deformed to more male and female targets, the generic models are shown in Appendix A and B.

Table 3: Summary of the comparison results with two approaches for the time and iterations needed to reach a same precision (root mean squared error = 0.006200m)

	RBF plus fine mapping (ours)	Multi-resolution Matching [4]
Time needed (minutes)	6.5	12.5
Iterations needed	160	600

We make a great improvement over the work from Allen et al. [4]. We started from repeating their work and found it was difficult to configure the three coefficients for the defined errors, i.e., the data, the marker and the smoothness errors. Rarely did we get any information from their work on how to decide the resolution for the lower-resolution generic model. Our analyses conclude that, applying their procedure to a new generic model will result in selection of different coefficients, decision of a different resolution for the lower-resolution generic model, and different number of iterations needed for acquiring good results. Therefore, our contribution by incorporating the RBF network lies in: we omit the marker error in error definitions, which makes easier the selection of coefficients; we no longer need to perform the alignment in a multi-resolution manner, which removes the effects from a sub sampled generic model of an improper resolution; and the whole process requires fewer error minimisation iterations to achieve similar minimal error, which results in a higher efficiency with the same-quality results.

3.2 Introduction of a new generic model

We also try the consistent parameterization with a new generic model. The original generic model is in a rectangular wireframe and has embedded eyeballs and teeth. Thus we start from removing the two parts and then perform a triangulation on the mesh.

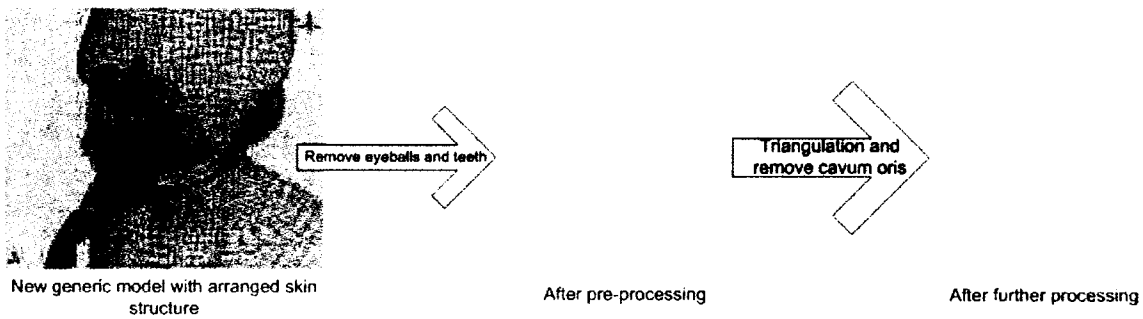


Figure 21: Pre-processing for the original generic model

Figure 21 shows the two-step processing on the original generic model. Since the original generic model is stored as a composition of four meshes, i.e., the whole body, the two eyeballs, and the teeth, we first remove the other three parts from the whole body. After this removal, we continue deleting the mesh of cavum-oris from the whole-body. This is useful in that we need such a generic model, which represents the shape of body surface but not of the inside structure. Therefore after deletion and triangulation with the aid of PolyWorks[1], a clean model is created.

As what we did on the previous generic model, we manually put 73 landmarks according to the definition of landmark positions in the CAESAR database. The new clean generic model with landmarks is shown in Figure 22. The processed generic model has 20,560 vertices and 41,009 triangles. Following the same parameterization procedure as what we did on the previous generic model, we get new experimental results as in Figure 23, where we map the new generic model to a Caucasian female and an African female. Figure 24 shows the wireframe of the resulting meshes, where the mesh structure

[1] PolyWorks, courtesy of InnovMetric Software Inc.

well represents the characteristic of the body surface, e.g., the breast and the navel. This mesh structure is very useful for body animation.

We try our approach on targets with different ethnicities. Appendix E and F show the parameterization results on Asian, Caucasian, and African males and females. Along each row, the first target is from the publicly available CAESAR dataset.

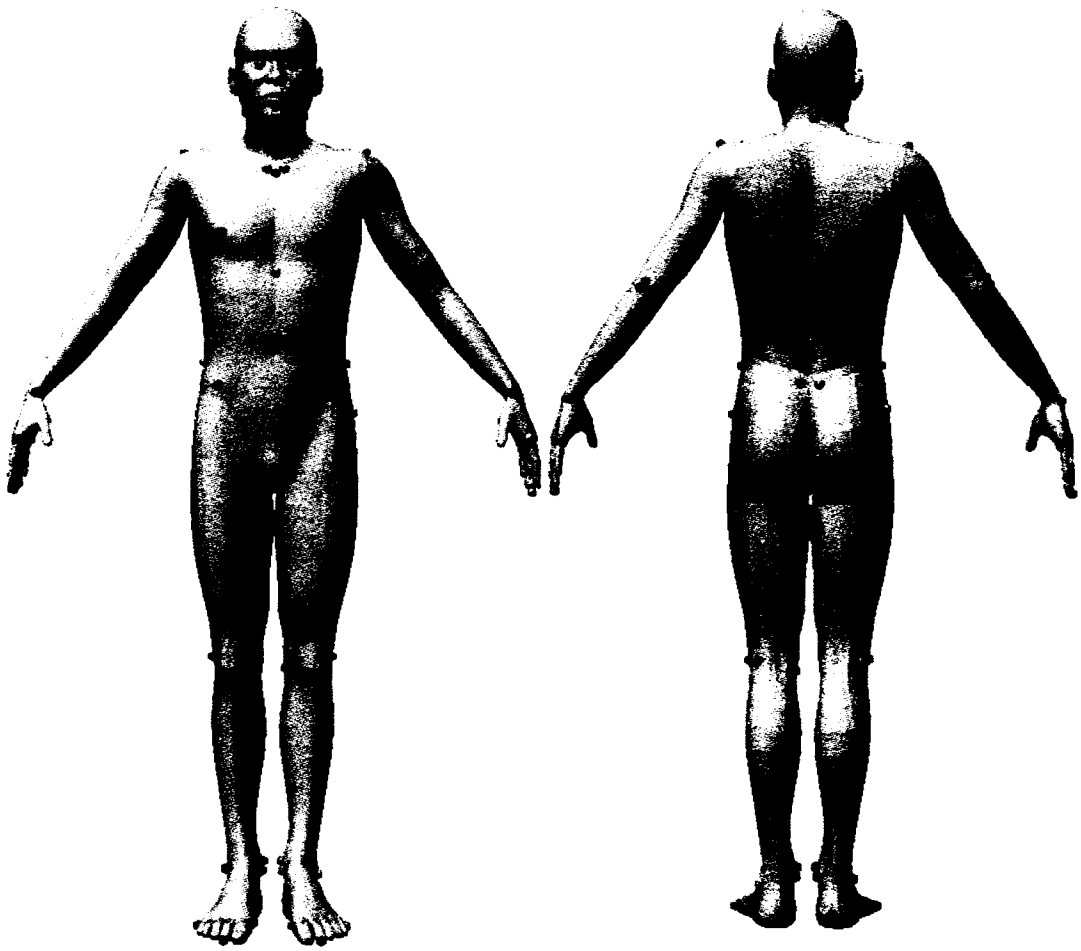


Figure 22: New clean generic model with 73 landmarks on its surface

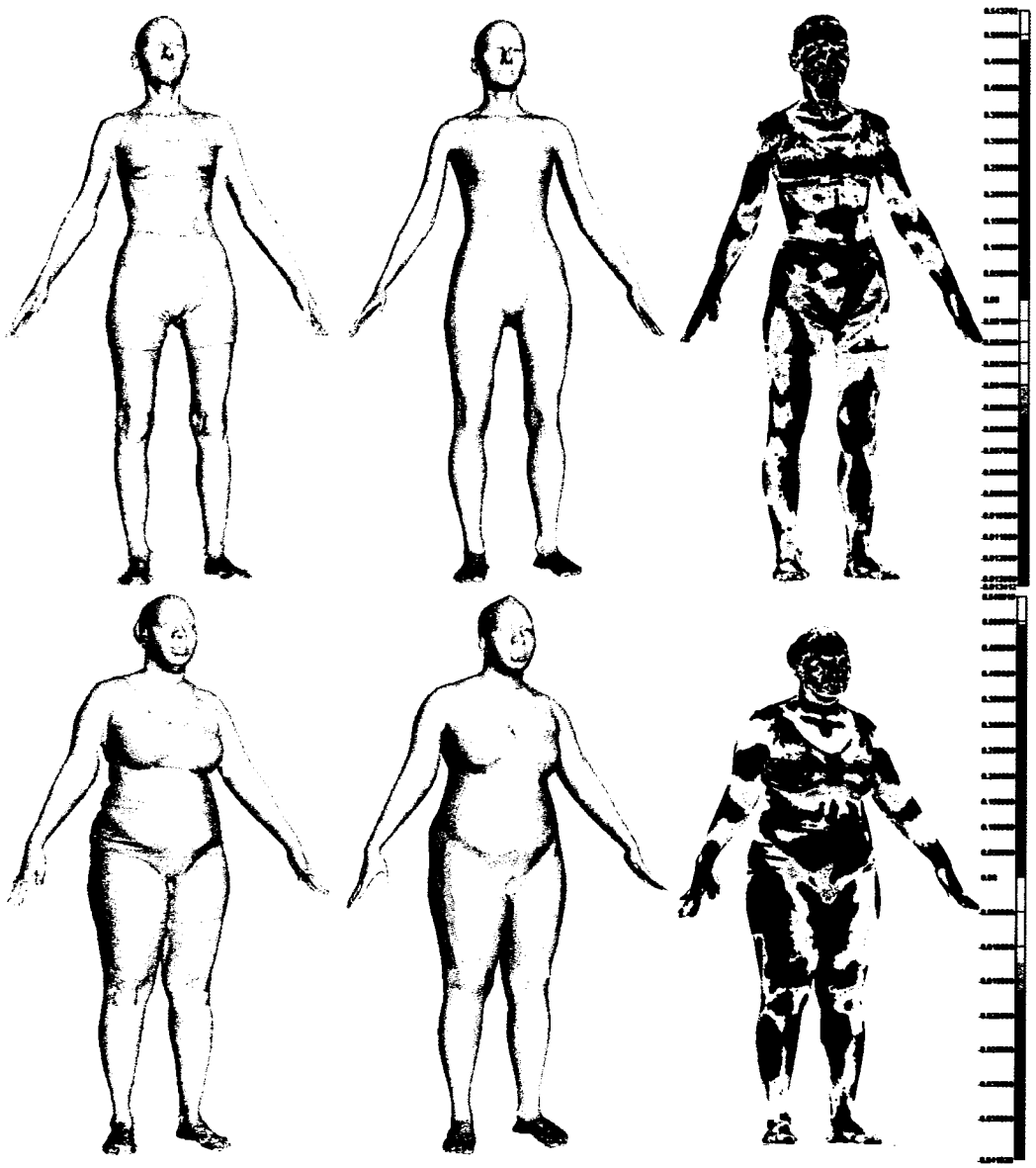


Figure 23: experimental results with the new generic model (upper row from left to right: the "csr0136a" in CAESAR database, the parameterized result, and a comparison in between; lower row from left to right: the "csr0213a" in database, the parameterized mesh and a comparison between two meshes)

We compare in Figure 25 the three generic models used in our project. The high-resolution generic models (with the 328K and 100K triangles) maintain detail surface

information after parameterization, while the generic model with an arranged structure creates nice mesh structure.

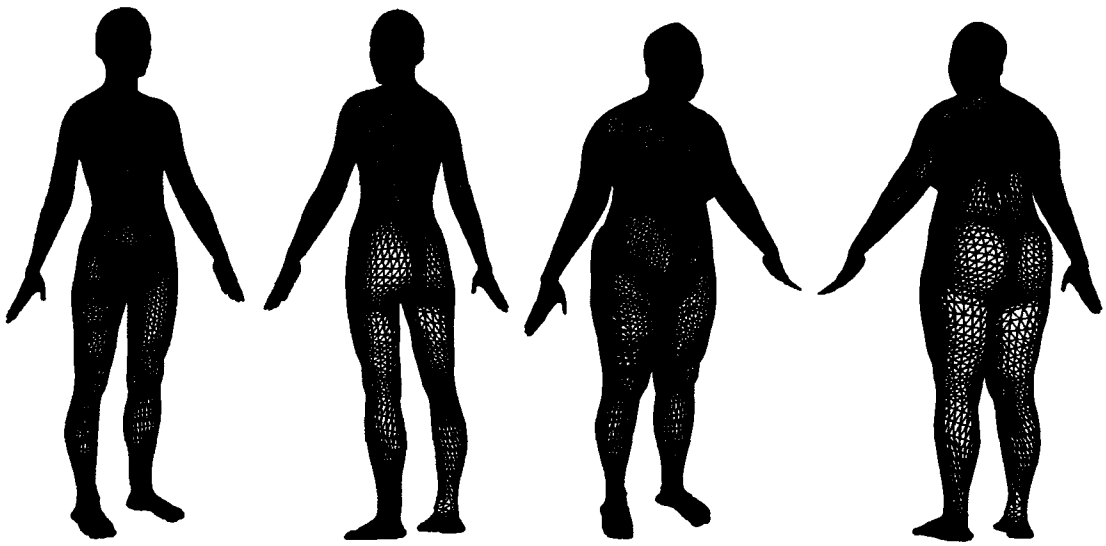


Figure 24: The mesh structure of the consistently represented body models (the left two are for "csr0136a", and the right two are for "csr0213a")

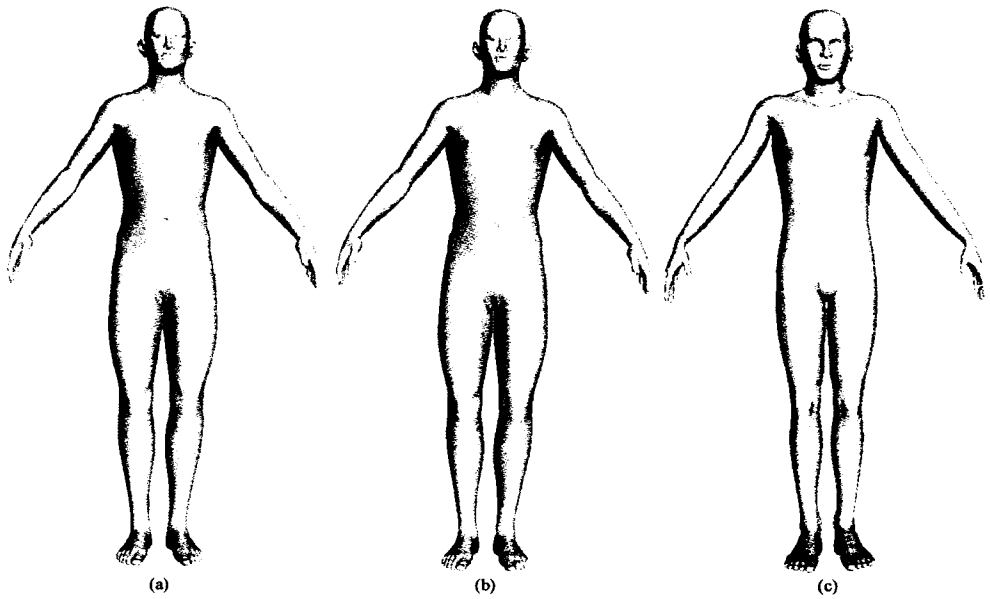


Figure 25: Visual comparison of the three generic models: (a) the generic model with 328K triangles, (b) the generic model with 100K triangles and (c) the generic model with arranged skin structure.

3.3 Consistent parameterization for 2D contours

To get a consistent representation of 2D contours, we start by using PolyWorks to get the silhouette of each 3D datum. We grab all the silhouettes using same setting for a perspective projection. Some examples are demonstrated in Figure 26.

We apply a toolkit in Matlab[2] software to trace all the contours of the silhouettes. By inputting coordinate of a starting point at the boundary, this tool automatically finds a sequence of points along the boundary and a sequenced string of these points makes a representation of the contour.

To get a well-corresponded representation (a consistent one) of all the contours, we also map 3D markers to find the features along the 2D boundary. Figure 27 shows some mapped markers from 3D landmarks (landmarks representing surface features in 3D still keep feature information in 2D), which are close to the boundary and can be used to find feature points along the contour for segmentation. Most of those feature points are found by searching for the closest points with a kd-tree, which is introduced in more detail later. Other feature points, e.g. 9^{th}, 16^{th} and 23^{rd} points, can be located by tracing the contour to find local ridges.

In computer science, a kd-tree (short for k-dimensional tree) is a space-partitioning data structure for organizing points in a k-dimensional space. Kd-trees are a useful data structure for several applications, such as searches involving a multidimensional search key (e.g. range searches and nearest neighbour searches). [34]

---

[2] Courtesy of the MathWorks, Inc. http://www.mathwork

A kd-tree uses only splitting planes that are perpendicular to one of the coordinate system axes. In addition, every node of a kd-tree, from the root to the leaves, stores a point. As a consequence, each splitting plane must go through one of the points in the kd-tree. [34]

The following code applies to how we can construct a kd-tree: [38]

```
function kdtree (list of points pointList, int depth)
{
  if pointList is empty
    return nil;
  else
  {
    // Select axis based on depth so that axis cycles through all valid values
    var int axis := depth mod k;

    // Sort point list and choose median as pivot element
    select median from pointList;

    // Create node and construct subtrees
    var tree_node node;
    node.location := median;
    node.leftChild := kdtree(points in pointList before median, depth+1);
    node.rightChild := kdtree(points in pointList after median, depth+1);
    return node;
  }
}
```

The nearest neighbour (NN) algorithm, to find the NN to a given target point not in the tree, relies on the ability to discard large portions of the tree by performing a simple test. To perform the NN calculation, the tree is searched in a depth-first fashion, refining the nearest distance. First the root node is examined with an initial assumption that the smallest distance to the next point is infinite. The subdomain (right or left), which is a hyperrectangle, containing the target point is searched. This is done recursively until a final minimum region containing the node is found. The algorithm then (through

recursion) examines each parent node, seeing if it is possible for the other domain to contain a point that is closer. This is performed by testing for the possibility of intersection between the hyper-rectangle and the hyper-sphere (formed by target node and current minimum radius). If the rectangle that has not been recursively examined yet does not intersect this sphere, then there is no way that the rectangle can contain a point that is a better nearest neighbour. This is repeated until all domains are either searched or discarded, thus leaving the nearest neighbour as the final result. In addition to this one also has the distance to the nearest neighbour on hand as well. [38]

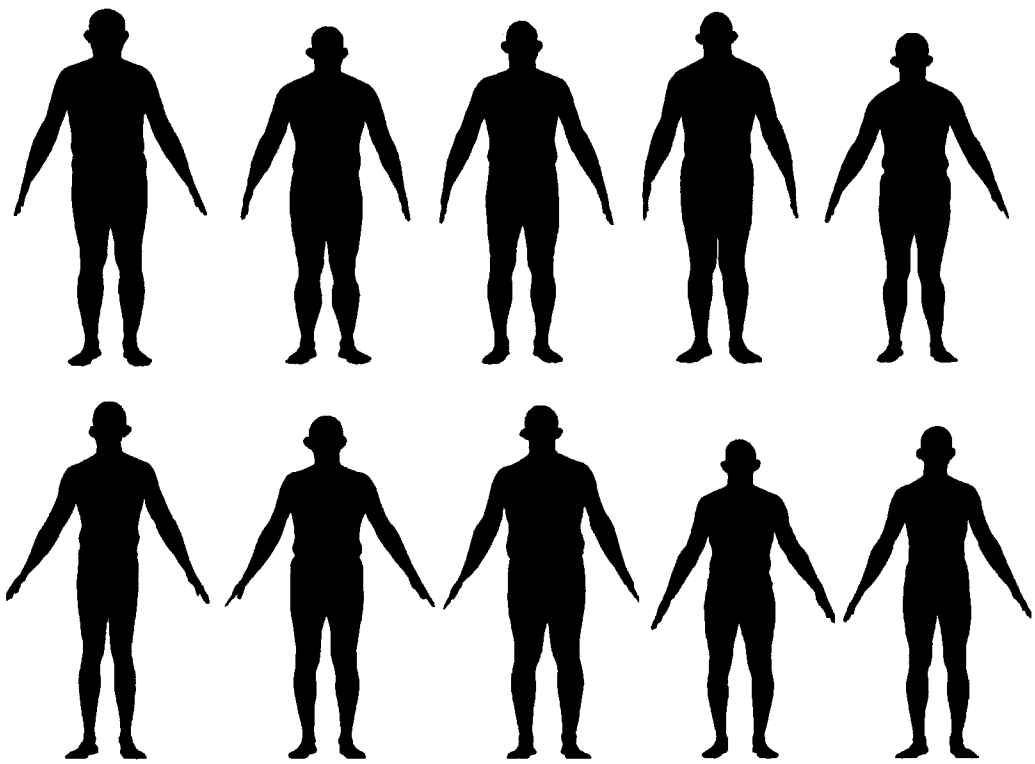


Figure 26: Sample silhouettes mapped from 3D data in a perspective view

In Figure 27, we also automatically locate three inner mapped markers, i.e. the 31st, 32nd and 33rd ones. These are the real mapped 2D locations of 3D markers hidden under

arms and between two legs. They are useful in representing anthropometric features of the 2D contour. Therefore, our work on contour representation also includes connecting these points and nearby feature points, which are already located out on the contour.

Thus the whole contour is divided into 36 neighbouring parts. This number is later reduced to be 30 (We represent contour part from feature point #8 to #31 without breaking at feature point #9. The next contour part starts from feature point #31 to #10 without breaking at #9. Similar processing was done in representing two contour parts meeting at feature point #32 and in representing two contour parts meeting at #33). Before the contour representations can be called consistent ones, we need to define a generic contour, which specifies numbers of points between neighbouring feature points (the number of pixels representing each segment is: 80, 60, 80, 70, 60, 50, 60, 60, 100, 40, 100, 120, 80, 100, 80, 80, 100, 80, 120, 100, 40, 100, 60, 60, 50, 60, 70, 80, 60, 80). For each instance contour, we calculate a ratio between the numbers of actual pixels in certain contour part to that of corresponding contour part in the generic model. This ratio is used to decide the sampling rate for consistently representing each consistent contour.

For testing images, we need to manually put 32 landmarks being close to contour features. The landmarks are outlined as light green points in each image as shown in Figure 28 (we hide subjects' faces because of identity reasons). The Matlab tool will find the contours for these subjects, which will be shown in the reconstruction part (Chapter 6). Using the landmarks in Figure 28, we are able to consistently represent the contours for each subject.

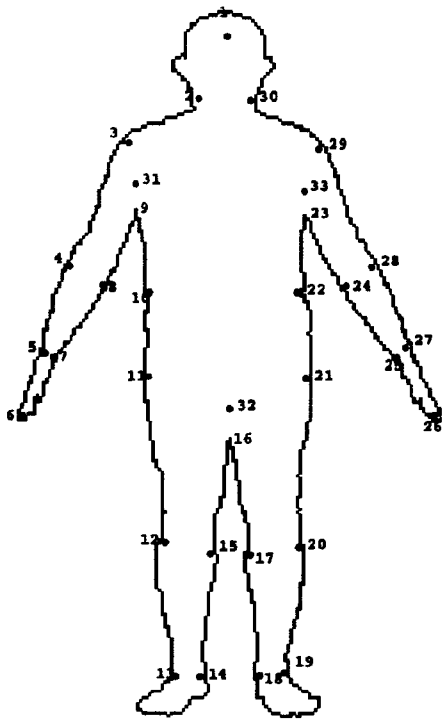


Figure 27: Contour and some of the mapped markers

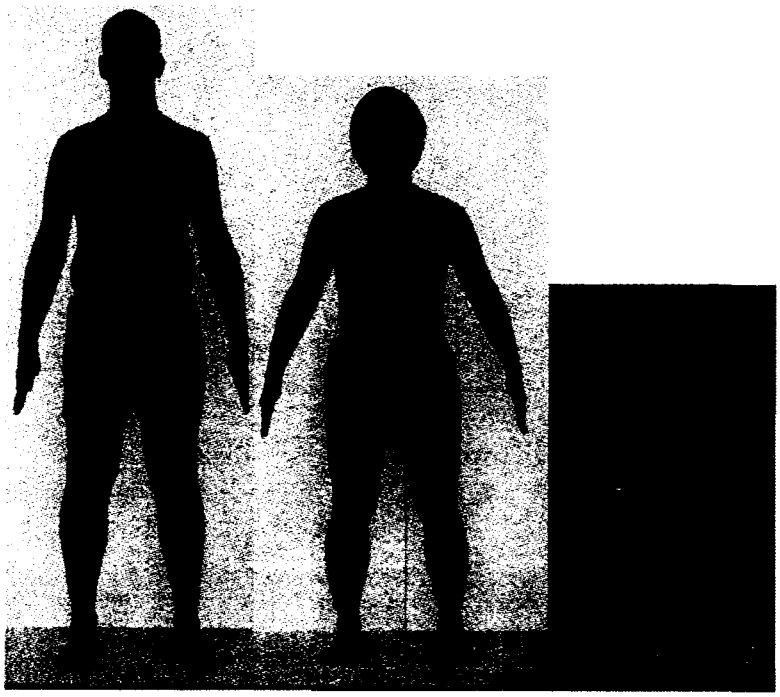


Figure 28: Markers manually put for consistent representation of 2D contours

Chapter 4: PCA on 3D Meshes and 2D Contours

In this chapter, Principal Components Analysis is applied to analyze the distribution of the human bodies and their segments in PCA spaces. We also conduct PCAs on the whole-body and segmented contours.

4.1 PCA analysis on the parameterized CAESAR data

4.1.1 Principal components analysis

Principal Components Analysis (PCA) is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension. It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data. [21]

The other main advantage of PCA is that, after finding these patterns in the data, and compressing the data by reducing the number of dimensions, we will not lose much information. [21] The general steps for conducting a PCA is as follows:

Step 1: Get some data

This is to collect the input data, which is to be analyzed by the PCA.

Step 2: Subtract the mean

For PCA to work properly, we need to subtract the mean from each of the data dimensions. This produces a data set whose mean is zero. [29]

Step 3: Calculate the covariance matrix

Covariance is such a measure. It is always measured between two dimensions. Calculating the covariance between one dimension and itself gets the variance. So, if we had a 3-dimensional data set (x, y, z), then we could measure the covariance between the x and y dimensions, the x and z dimensions, and the y and z dimensions. Measuring the covariance between x and x, or y and y, or z and z would give the variance of the x, y and z dimensions respectively. [21]

If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated. A useful way to get all the possible covariance values between all the different dimensions is to calculate them all and put them in a matrix. [21]

Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix

The set of eigenvectors x for A is defined as those vectors which, when multiplied by A, result in a simple scaling λ of x. Thus, $Ax=\lambda x$. Multiplying the right hand side by an identity matrix I, we have therefore $(A-\lambda I)x=0$. Solution to this equation will give the eigenvectors and eigenvalues for the A.

Eigenvectors can only be found for square matrices. Not every square matrix has eigenvectors. And, given an $n \times n$ matrix that does have eigenvectors, there are n of them.

Another property of eigenvectors is that even if we scale the vector by some amount before we multiply it, we still get the same multiple of it as a result. All the eigenvectors of a matrix are orthogonal, i.e., at right angles to each other, no matter how many dimensions we have. [21]

A Matlab code for step 1 to 4 is also pasted from our project:

```
% step 1, load the data
load consis_female.mat;

% step 2, subtract the mean
dim_cons = size(CONSISTENT_MATRIX);
gave = zeros(1,dim_cons(2));
texno = dim_cons(2);
for i=1:dim_cons(1)
   gave(1:texno)=gave(1:texno)+CONSISTENT_MATRIX(i,1:texno);
end;
gave=gave/dim_cons(1);
X=zeros(dim_cons(1),texno);
for i=1:dim_cons(1)
   X(i,1:texno)=CONSISTENT_MATRIX(i,1:texno)-gave(1:texno);
end;

% step 3, calculate the covariance matrix
R=X'*X;

% step 4, calculate the eigenvectors and eigenvalues
[Y1, D1] = eig(R, 'nobalance');
```

When we find eigenvectors, we like to find the eigenvectors whose length is exactly one. This is because, the length of a vector does not affect whether it is an eigenvector or not, whereas the direction does. So, in order to keep eigenvectors standard, we usually scale it to make it have a length of 1, so that all eigenvectors have the same length. Eigenvalues are closely related with eigenvectors. They always come in pairs. [21]

By this process of taking the eigenvectors of the covariance matrix, we have been able to extract lines that characterise the data. The rest of the steps involve transforming the data so that it is expressed in terms of eigenvectors. [21]

Step 5: Choosing components and forming a feature vector

Here is where the notion of data compression and reduced dimensionality comes into it. The eigenvector with the highest eigenvalue is the principal component of the data set. In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives us the components in order of significance. Now, we can decide to ignore the components of lesser significance. We do lose some information, but if the eigenvalues are small, we do not lose much. [21]

What needs to be done now is to form a feature vector, which is constructed by taking the eigenvectors that we want to keep from the list of eigenvectors, and forming a matrix with these eigenvectors in the columns. [21]

Step 6: Deriving the new data set

Once we have chosen the components that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the original data set, transposed: *FinalData* = *RowFeatureVector* × *RowDataAdjust*, where *RowFeatureVector* is the matrix with the eigenvectors in the columns transposed so that the eigenvectors are now in the rows, with the most significant eigenvector at the top, and *RowDataAdjust* is the mean-adjusted data transposed, i.e., the data items are in each column, with each row holding a separate dimension. *FinalData* is the final data set, with data items in columns, and dimensions along rows. [21]

This will give us the original data solely in terms of the vectors we choose. It is possible to express data in terms of any axes that we like. We have transformed our data so that they are expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data. This is helpful because we have now classified our data point as a combination of the contributions from each of those lines. [21]

How do we get the original data back? Before we do that, remember that only if we took all the eigenvectors in our transformation will we get exactly the original data back. If we have reduced the number of eigenvectors in the final transformation, then the retrieved data has lost some information. The previous formula on deriving new data set can be turned around so that, to get the original data back, *RowDataAdjust* = *RowFeatureVector* $^{-1}$ × *FinalData*, where *RowFeatureVector* $^{-1}$ is the inverse of *RowFeatureVector*. However, when we take all the eigenvectors in our feature vector, it turns out that the inverse of our feature vector is actually equal to the transpose of our feature vector. This is only true because the elements of the matrix are all the unit eigenvectors of our data set. Thus we have *RowDataAdjust* = *RowFeatureVector* ^{T} × *FinalData.* But to get the actual data back, we need to add on the mean of that original data: *RowDataAdjust* = (*RowFeatureVector* ^{T} × *FinalData)* + *OriginalMean.* [21]

In Figure 29, after conducting an eigen-decomposition on all the data, we find two orthogonal vectors, i.e. principal component I (PC1) and principal component II (PC2). We observe that the main variation in the dataset is distributed along PC1, while a lesser obvious one lies along PC2. Therefore, data X can be approximated by $X' = \overline{X} + b \cdot PC_1$.

We used this 2D example to introduce the PCA and it also applies to a dataset in a higher dimension. [29]

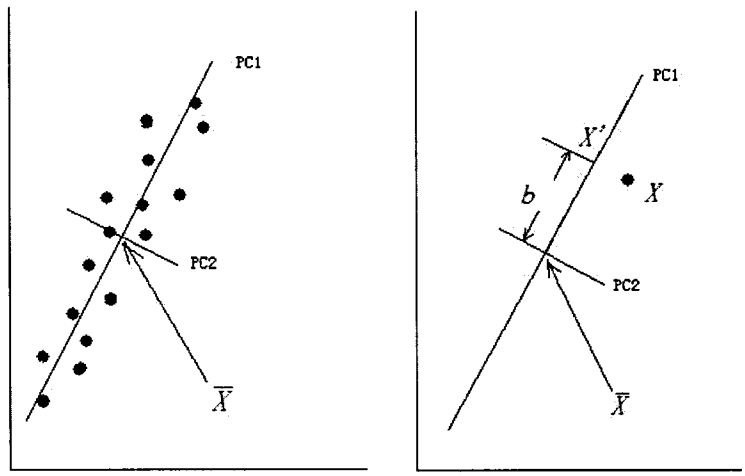


Figure 29: Principal components analysis

4.1.2 Pattern variation of CAESAR dataset

The parameterized representation of each data in the CAESAR has a well-corresponded same number of vertices. By putting coordinates of all the vertices into a vector S_i, we can have a dataset $\{S_i\}$, $i=1...m$, where m is the number of 3D data in the database. Then we calculate the average of all the data: $\overline{S} = \frac{1}{m}\sum_{i=1}^{m} S_i$, and a covariance matrix $Cov = \frac{1}{m-1}\sum_{i=1}^{m}(S_i - \overline{S})(S_i - \overline{S})^T$. Conducting an eigen-decomposition on this covariance matrix yields a set of eigen-vectors and eigen-values. The eigen-vectors with large eigen-values always represent the biggest variation among the dataset.

According to the PCA theory, new 3D human body can be reconstructed from:

$$S = \overline{S} + \Phi \cdot B, \tag{9}$$

where Φ is the eigen-vectors corresponding to a set of descending eigen-values. B is a variable in column vector, in which each cell value can be changed to reconstruct

different 3D data based on the PCA-space. Setting all the cells in vector B to be zero, the reconstructed 3D data will be an average model of all the 3D human bodies.

Based on the PCA theory, we expect that changing the first cell in B will create a most obvious variance among the database. Figure 30 shows an obvious change in height.

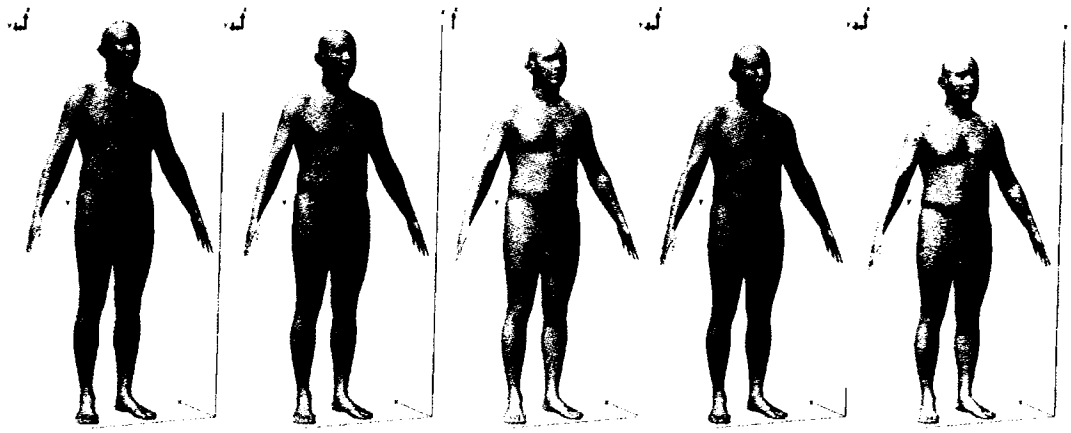


Figure 30: First pattern variation among the database: the height (from left to right: B_1=-20, -10, 0, 10, 20. Other cells in B are all set to be zero)

By changing the second cell value in B, a sequence of reconstructed 3D bodies is shown in Figure 31, where the most change lies in body weight. We must add that in each pattern variation, there will not be a single difference among all the reconstructed models. Here arm pose change is small compared with the weight variation.

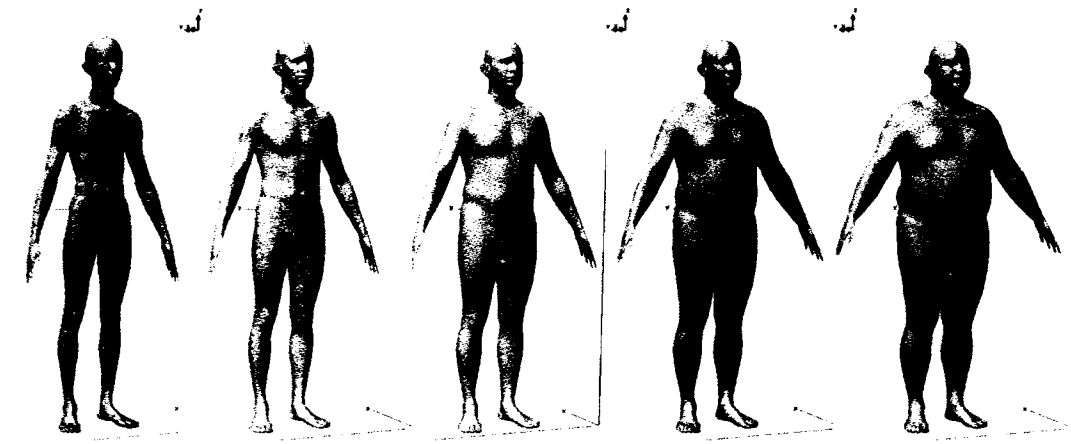


Figure 31: Second pattern variation among the database: the weight and arm pose (from left to right: B_2=-20, -10, 0, 10, 20. Other cells in B are all set to be zero)

Pose changes can be observed from sample data in Appendix A and B. By increasing the value of B_3, there is a change in the body pose from backward to forward as shown in Figure 32. More pattern variations are demonstrated in Appendix C. Appendix D also presents our PCA on a database of 728 male and female bodies (approximately half of all the data are female bodies). Our results are consistent to those presented in [37].

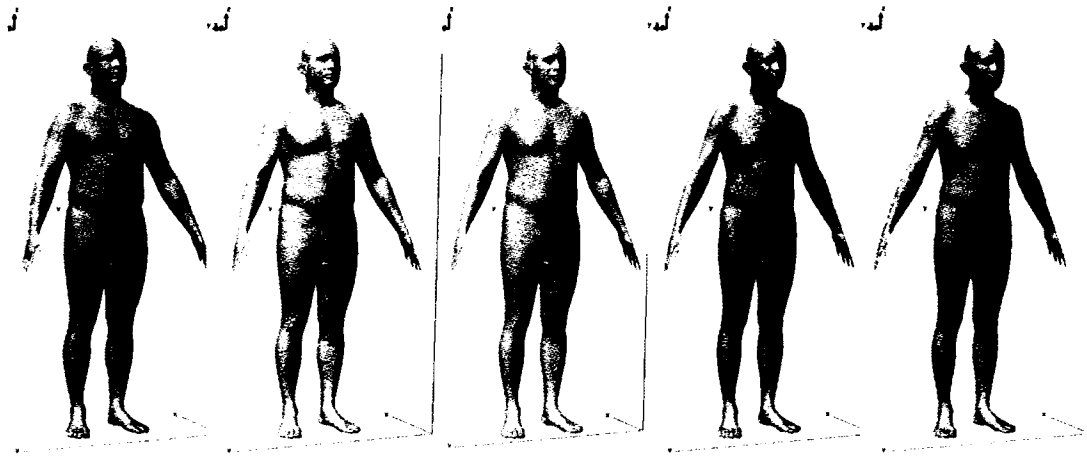


Figure 32: Third pattern variation among the database: pose from backward to forward (from left to right: B_3=-16, -8, 0, 8, 16. Other cells in B are all set to be zero)

4.2 PCA analysis on Segmented CAESAR data

Our original idea was to learn the deformation of each human body segment and apply this variation to constructing human body segments. Therefore, we discuss in this section the segmentation on the generic model and PCA analysis on variations of each segment.

4.2.1 Segmentation on the generic model

We separate the generic model into six parts: head, left arm, right arm, torso, left and right legs. Neighbouring segments have an overlapping, which is basically for stitching work later. Figure 33 displays segmented generic models with and without the torso.

After segmentation on the generic model, we are able to find for each segment index files in the whole generic model. This is useful in several aspects: they help find the

segments for each aligned CAESAR data in a consistent way, help find the overlapping band patches between two neighbouring segments, and help merge segments into a complete result model.

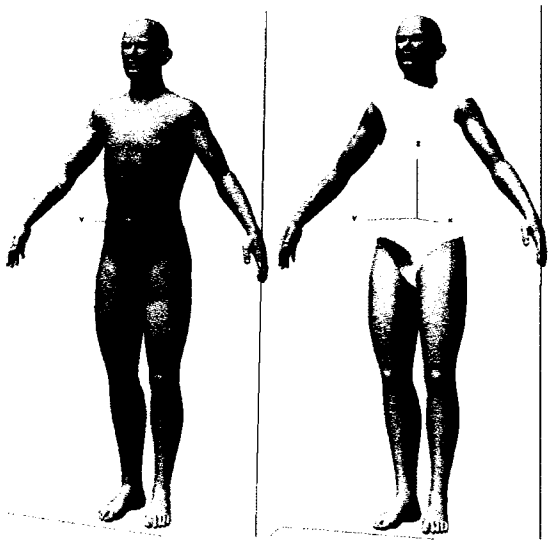


Figure 33: segmented generic model (with and without the torso)

4.2.2 PCA analysis on segmented human body models

Since each segment is part of the whole body, we expect to see some similar changes as those from the whole body. For human head, we find the first three pattern variations correspond mainly to the height, pose change along forward-backward and left-right directions. After removing these changes, we also find face shape changes in Figure 34, where the middle column pictures are the average head. These pictures again support our statement that along each eigen-vector, there is always a combination of changes.

PCA analysis on the torso also creates interesting results as shown in Figure 35. Besides the first pattern change in height, the second variation lies in weight, and the

third pattern corresponds to a volume-preserving deformation. The next few pattern changes relate to the pose change and shape deformation, e.g. bending of the torso.

As Figure 36 shows, the first pattern variation controls the height of the left arm, while other four patterns correspond to different pose variations. The similar variation on height also exists in the first pattern of changes for right leg in Figure 37. Figure 38 displays the other four changes corresponding mainly to pose and scale of the right leg.

All these images display a richer variation of body segments than that of the whole body. Therefore, based on PCA-space for each body segment, we will be able to better reconstruct the whole body based on segmented human bodies.

We conclude that there is a reasonable interval of [-20, 20] for changing the weight along each eigen-vector and this interval is shrank while the eigen-value of corresponding eigen-vector decreases (the selection of weight to be 20 or –20 would create too extreme results along eigen-vectors with smaller eigen-values).

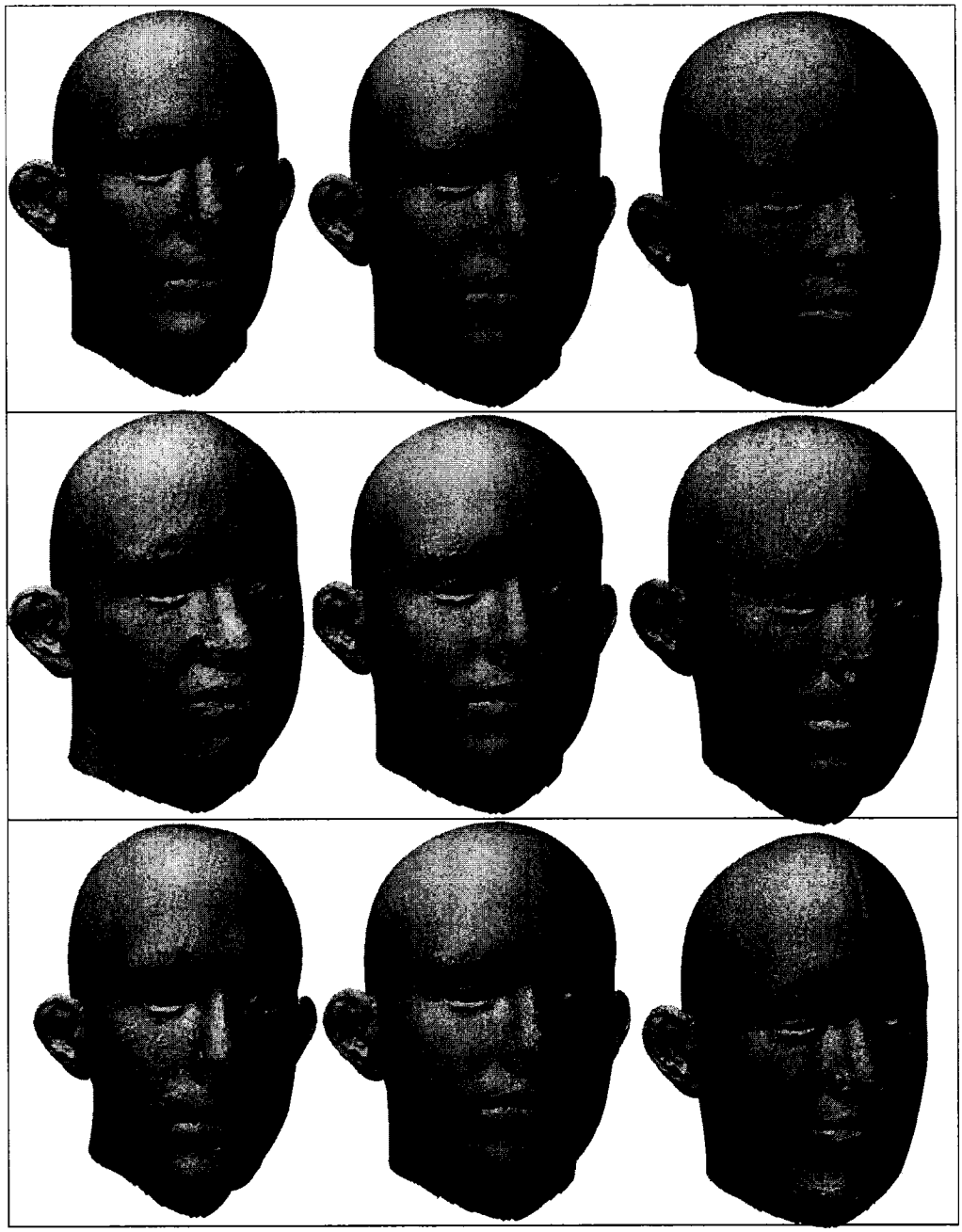


Figure 34: First three pattern variations of head shape besides the changes in pose (first row also corresponds to height change, second row also corresponds forward-backward pose change, and third row also has a change in pose along left-right direction)

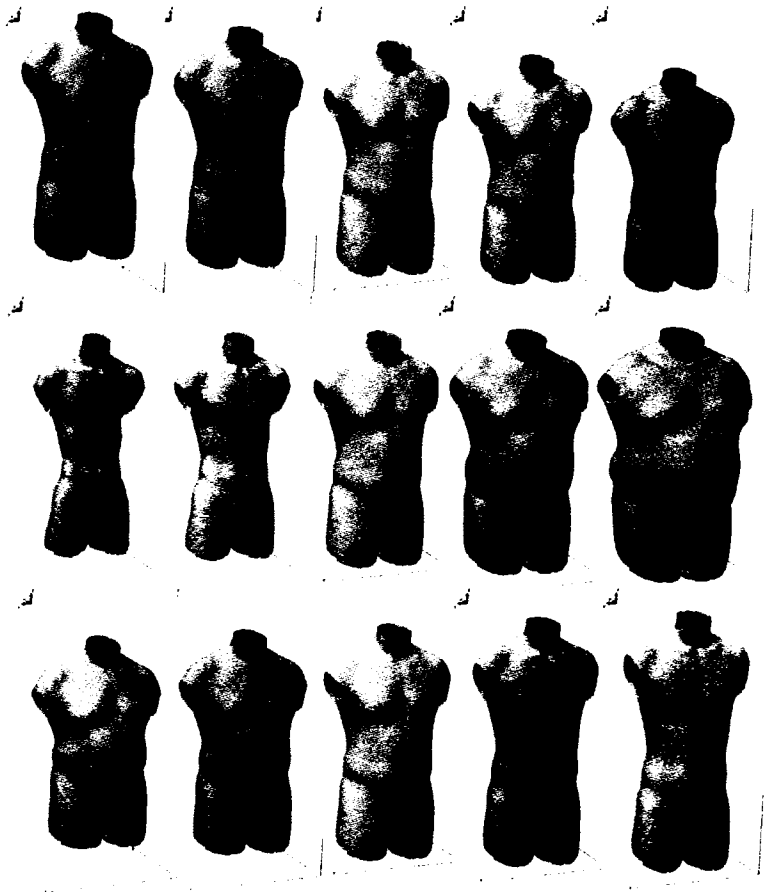


Figure 35: First three pattern variations of the torso (the middle column torsos are the average torso)

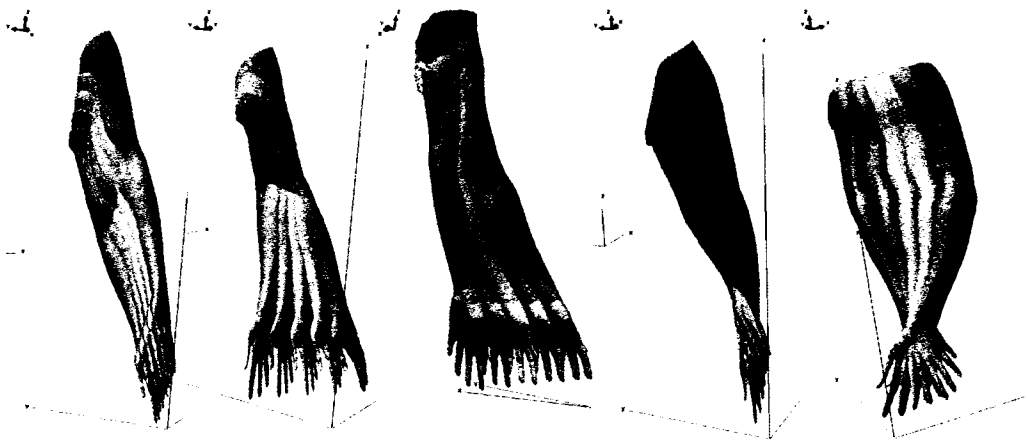


Figure 36: First five pattern variations of left arm

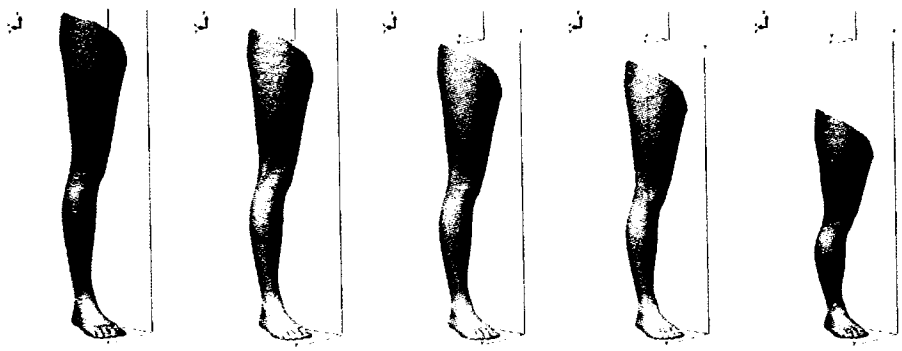


Figure 37: First pattern change in right leg corresponds to height

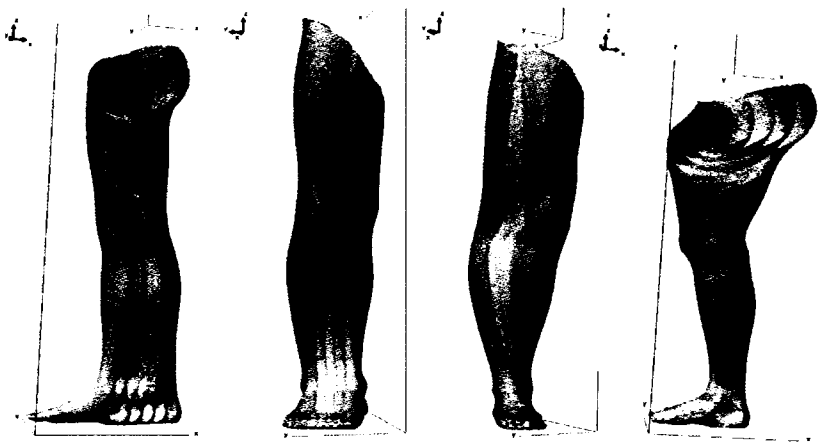


Figure 38: 2nd to 4th pattern variations of the right leg

4.3 PCA analysis on 2D contours

Based on a consistent representation of the contour for each silhouette, we are able to conduct a PCA and thus find the variations along eigen-vectors.

After selecting different weights along eigen-vectors, we thus get the variations of the contour in Figure 39. Similar to those results in 3D analysis, the first principal component corresponds to height of the mapped 2D contour. Second variation represents a change in scale. The next six variations are mainly on weight and pose.

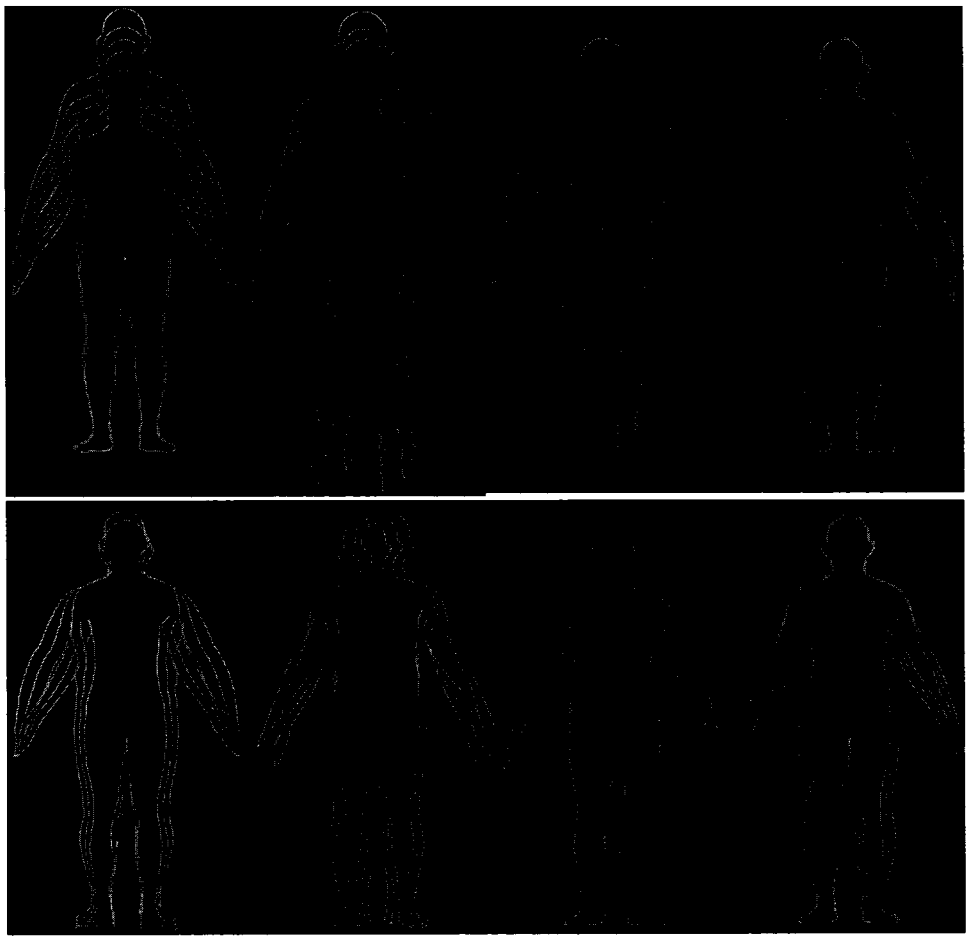


Figure 39: PCA analysis on 2D contours (upper row from left to right corresponds to the first four principal components, lower row from left to right relates to the next four principal components)

Comparing these variations with those from 3D whole body, we find some common variations. The variations found in 2D are a subset of those in 3D in that previous ones do not have the pose changes along backward-forward directions. This observation presents an experimental basis for finding a relationship between the 2D and 3D PCA-spaces.

Even though it is possible to find contour for each 3D human body part by mapping them to 2D, we directly find contour parts from the whole-body contour in reference to mapped landmarks. This is to keep the completeness of the whole contour and to make

full use of 3D markers' information. Since there are overlapping between neighbouring body parts, there will also be overlapping between neighbouring contour parts. We solve this problem by tracing backward and forward some pixels from certain feature points (for example, we trace forward 20 pixels in counter-clockwise direction from feature point #11 to represent the lower-left end of the torso's contour for each body).

Besides the results on whole-body contour PCA, we also conduct a series of experiments on contour parts as shown from Figure 40 to Figure 43. Besides similar variations, we also find some variations, which are not available in the whole-body PCA results. Segmented 2D PCA presents a more variable space than the whole-body PCA.

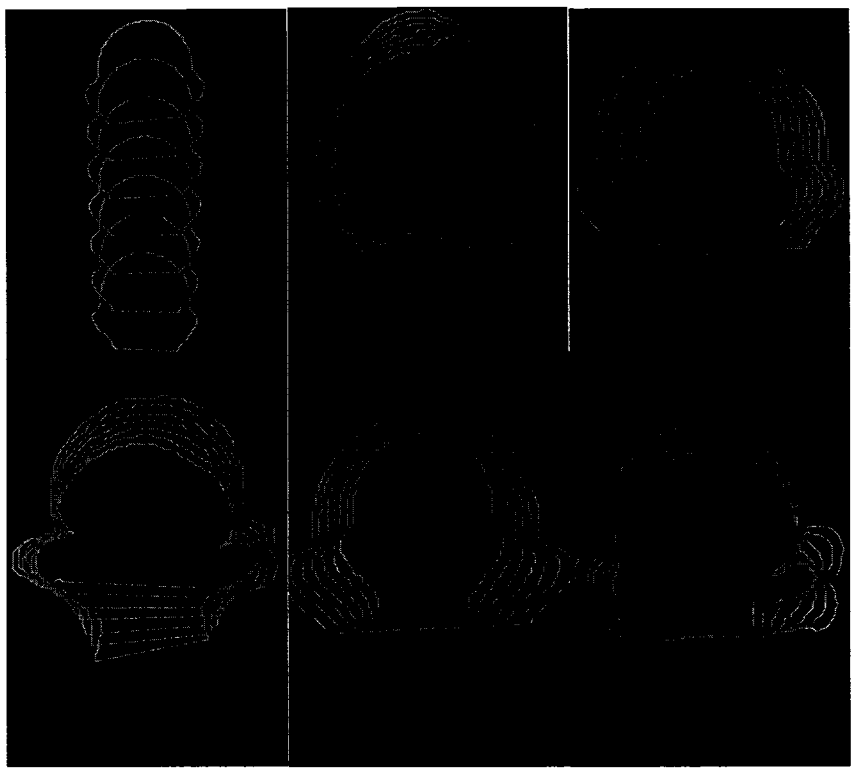


Figure 40: PCA analysis on 2D head contour (upper row from left to right corresponds to the first three principal components, lower row from left to right relates to the next three principal components)

In Figure 40, the first pattern corresponds to the height of head, the second one represents a rotation on the head contour, the third one is on a left-right translation and the fifth one represents a scaling variation. The fourth and sixth patterns represent other variations.

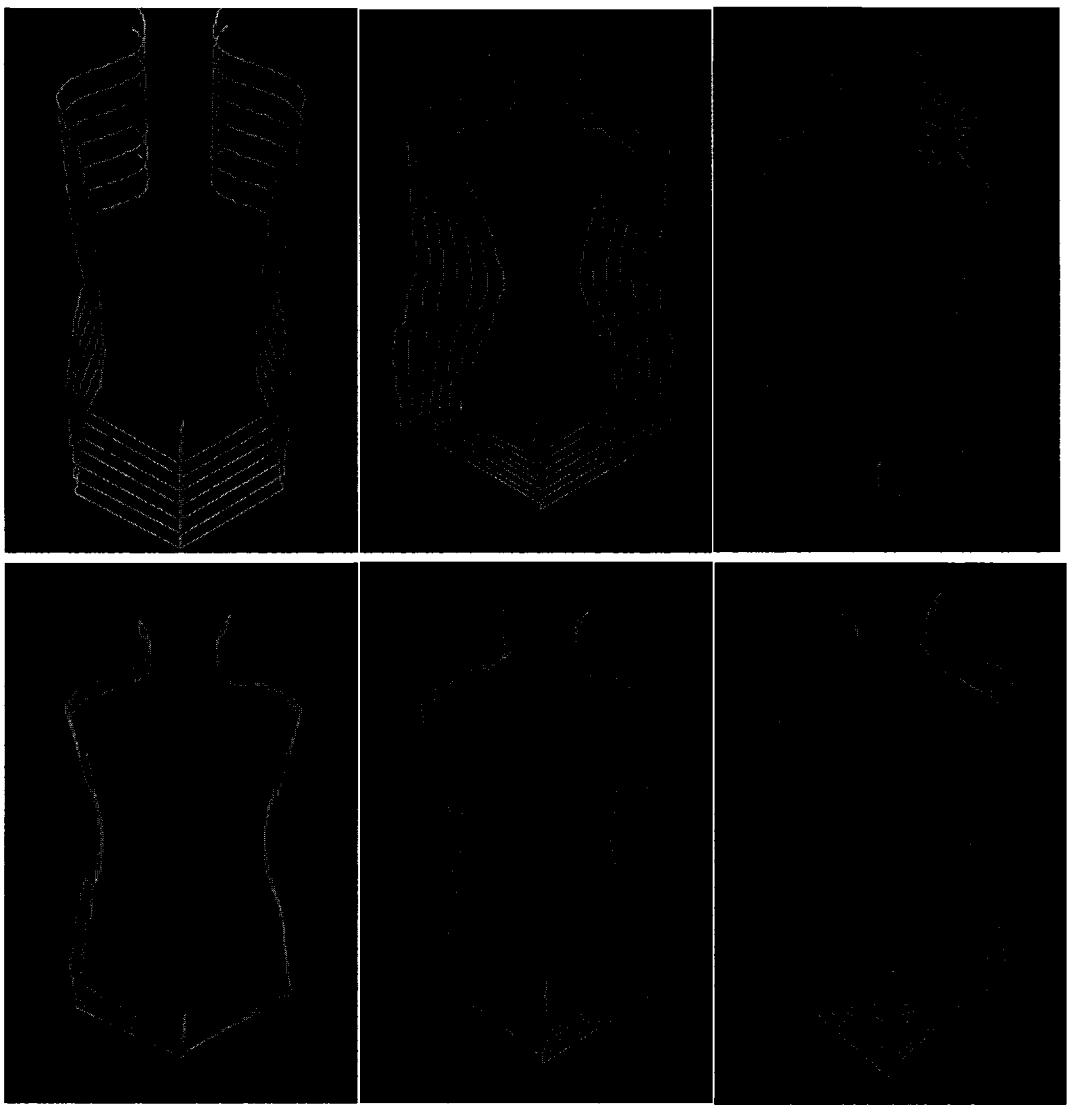


Figure 41: PCA analysis on 2D torso contour (upper row from left to right corresponds to the first three principal components, lower row from left to right relates to the next three principal components)

In Figure 41, the first pattern corresponds to the height of torso, the second one represents a change of contour weight, the sixth one is on the height of the lower boundary, and the third, fourth and fifth patterns represent other variations.

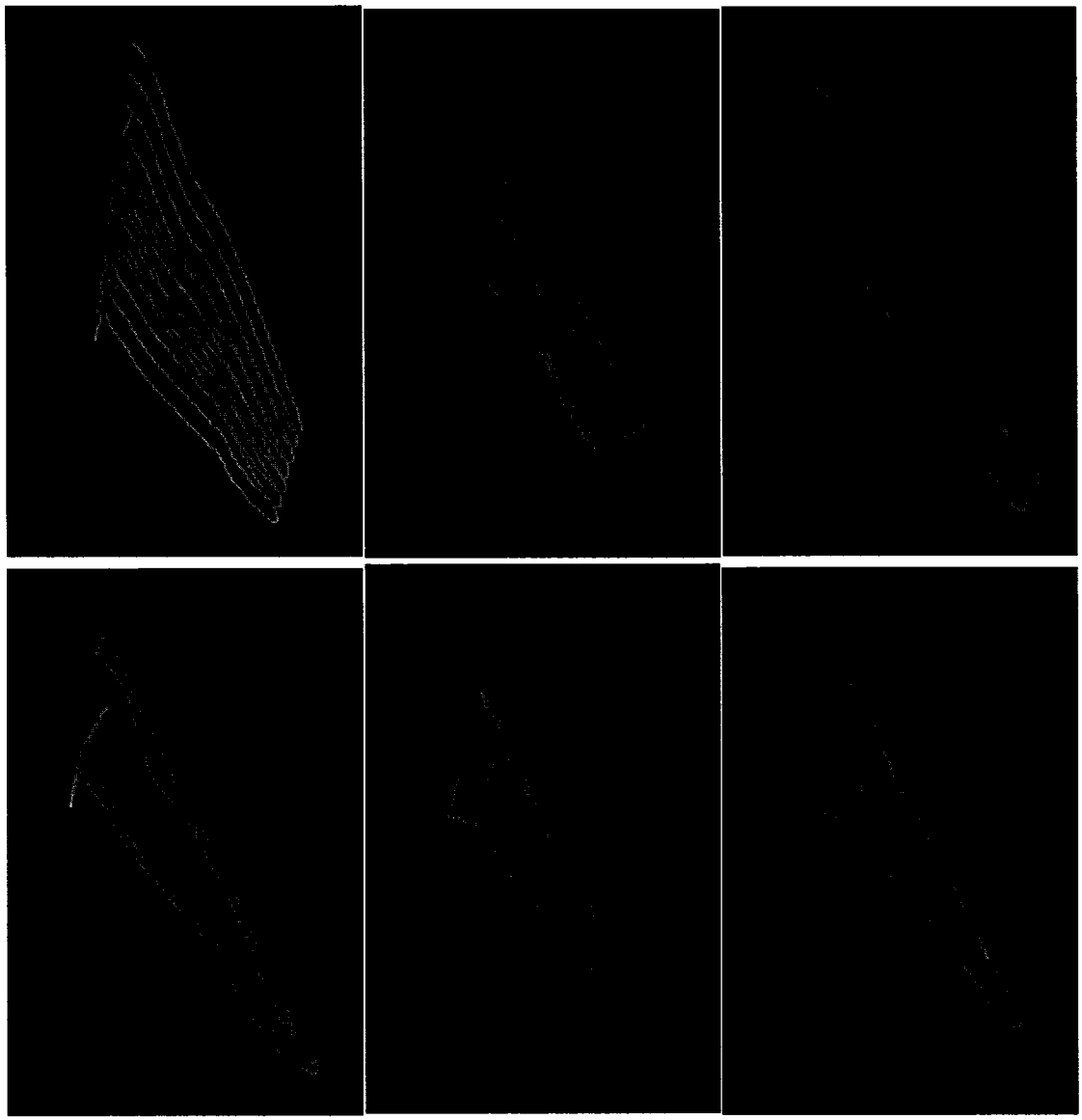


Figure 42: PCA analysis on 2D left-arm contour (upper row from left to right corresponds to the first three principal components, lower row from left to right relates to the next three principal components)

In Figure 42, the first pattern corresponds to the height of left arm, the second one represents a rotation from the upper-end of the arm, and the third one is on a rotation from the middle of the arm. The fourth, fifth and sixth patterns represent other variations.

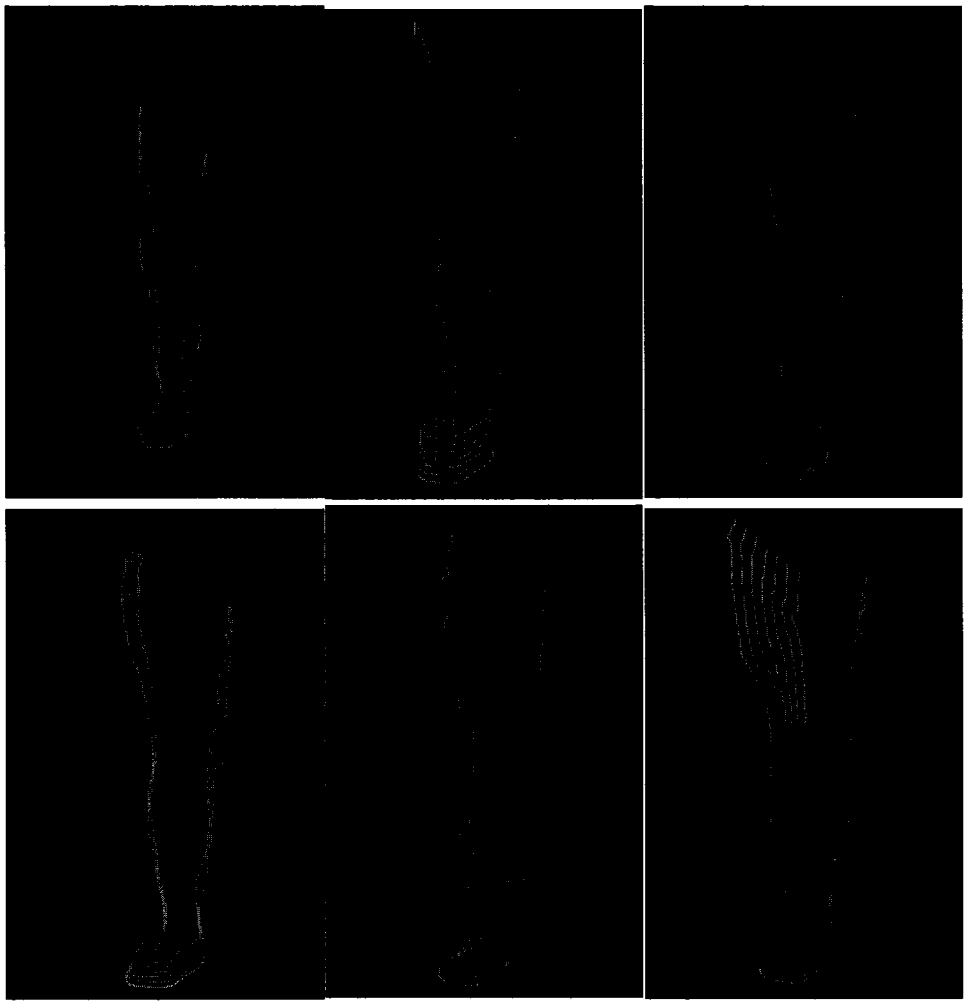


Figure 43: PCA analysis on 2D right-leg contour (upper row from left to right corresponds to the first three principal components, lower row from left to right relates to the next three principal components)

In Figure 43, the first pattern corresponds to the height of right leg and the sixth one represents a weight difference. The second, third, fourth and fifth patterns represent other variations.

Chapter 5: Relationship Learning from 2D to 3D PCA-spaces

After building PCA-spaces for 3D whole body and segmentations as well as those of 2D contours and segmentations, we are able to find a relationship between the 3D and 2D spaces. Taking an image of a new person and using its contour as an input, we can find its coordinate in the 2D contour PCA-spaces (whole body and segmented PCA-spaces). By mapping this coordinate back to the 3D PCA-space, we are able to reconstruct its 3D whole body or 3D segmented body parts with the 3D eigen-vectors.

5.1 Training for a relationship between 2D and 3D PCA spaces

In Chapter 4, we calculate and save the eigen-vectors for 3D human bodies and 2D silhouette contours. We also found good basis to our assumption that there exists a relation between the 2D and 3D PCA-spaces. Since eigen-vectors in 3D have a significantly higher dimension than those in 2D, it is very difficult to find a direct relationship through eigen-vectors. Even if we could find one, this mapping would have big errors in mapping from the 2D PCA-space to the 3D PCA-space.

We introduce in Chapter 4 that the pattern variations of 2D contours are a subset of those of 3D bodies; therefore, we are going to find a simple relationship, assuming a linear one, between the two PCA-spaces. We also notice that a coordinate vector in 3D

PCA-space is the final goal for whole body reconstruction; therefore, we start the training process by mapping each 3D datum onto the principal components in 3D PCA-space. Here the number of principal components is controllable.

By mapping each 3D datum onto the first k_1 principal components and forming the mapped coordinate vectors into a whole matrix, we have a (k_1×m) coefficient matrix C_1, where m is the number of human body scans used for training. Similarly, we can get another coefficient matrix C_2 with a dimension of (k_2×m), where k_2 is the number of principal components maintained for whole body 2D contour PCA-space.

Suppose there is a linear relationship D between C_1 and C_2:

$$C_1 = C_2 \times D, \tag{10}$$

then D can be calculated by solving a minimum mean-squared error problem (MMSE). The expression of D is as:

$$D = C_2 \cdot (C_2^T \cdot C_2)^{-1} \cdot C_2^T \cdot C_1, \tag{11}$$

This matrix D is thus a bridge between 3D and 2D eigen-spaces, where eigen-vectors in each space are bases to this bridge.

The above relation is built on the 3D human body PCA-space and the 2D whole body contour PCA-space. In a similar way, another six similar relationships can be built between 3D human body segments and their 2D contours.

5.2 Testing: Coordinate acquisition and mapping

5.2.1 Acquisition of an aligned contour for the new image

For a new image, we need to perform a translation and scaling transformation process on its contour so that it could become compatible to those contours in the training dataset. Then we can map this contour into the 2D contour PCA-space for further calculations. Figure 44 displays a testing image of a subject, its silhouette, and an outlined contour in green. This subject is not inside those bodies scanned for CAESAR project. The original image was taken by an Olympus D580 camera with flash on at a distance of 5.2m.

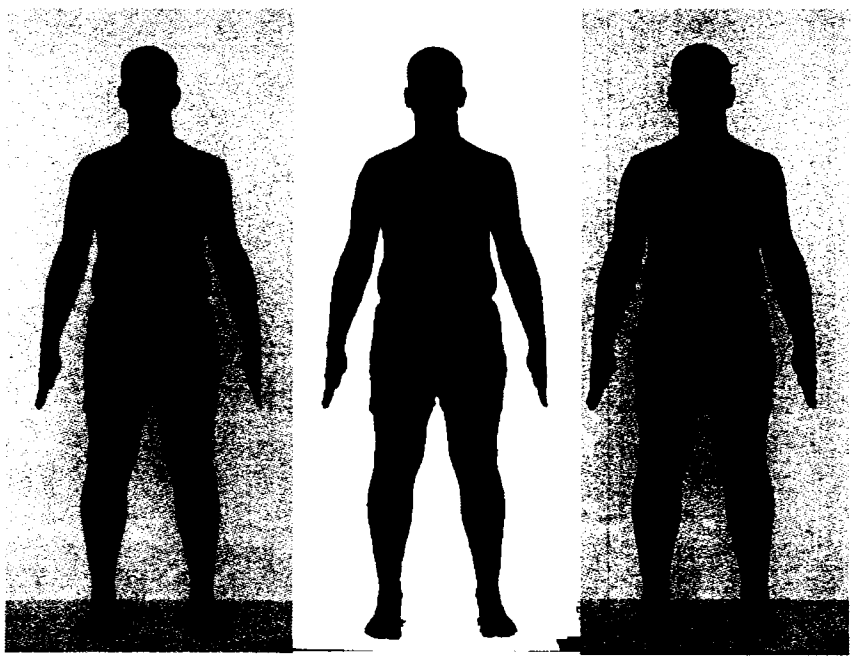


Figure 44: Contour finding for a new image (from left to right: the new subject, its silhouette, and its contour represented in green)

After launching the Matlab tool for contour finding, we manually put some markers for this new subject (the location of landmarks are introduced in section 3.3). This

manual work helps represent the new contour in compatibility with those training contours.

5.2.2 Coordinate mapping

After mapping this new whole body contour into the 2D PCA-space, we get a coordinate vector denoted as c_2, which has a dimension of the number of principal components maintained in 2D. Applying the relationship defined in (10), we are able to calculate the mapped 3D coordinate vector c_1 by:

$$c_1 = c_2 \times D. \tag{12}$$

The results c_1 has a dimension of the number of principal components we maintain in 3D PCA-space. A weighted summation of the selected principal components with this mapped weight vector will create the reconstruction result.

Selection of a different number of principal components in 2D or 3D PCA-spaces needs be done in the training process when we map each datum onto the eigen-vectors. Then the dimension of the matrix D will be consistent to the number of principal components selected (the number of rows equals the number of principal components selected in 3D, and the number of columns will be that of principal components maintained in 2D).

Chapter 6: Reconstruction of a 3D human body with 2D contours

We apply the learned information, a linear relationship between 2D and 3D PCA-spaces, to reconstructing a new subject's 3D body model through whole-body and segmented reconstructions. For the segmented reconstruction, we also discuss how to stitch the body parts into a complete and smooth body model.

6.1 Reconstruction with the whole-body contour

After getting the coordinate vector c_1, we can reconstruct the whole body by calculating a weighted sum of the principal components in 3D.

Figure 45 shows in the left a comparison between original contour and a reconstructed contour with first 50 principal components. Errors between the two contours contribute to the error in reconstructed 3D datum as shown in the right two images. Body shape of the subject selected here belongs to the distribution of that of the training database except that the arm pose is slightly different from those in the database. Therefore, some artefacts on both hands appear in Figure 45.

We also compare the reconstruction results by selecting different number of 2D and 3D principal components. Increment of 2D components makes the body more bent as shown in Figure 46. This effect also appears in another reconstruction example for

reconstructing a 3D body for an Asian subject. It seems to be reasonable for getting a better-reconstructed 3D datum by increasing the number of 3D components; however, experiments give even worse data: the surface becomes coarser and less human-alike.

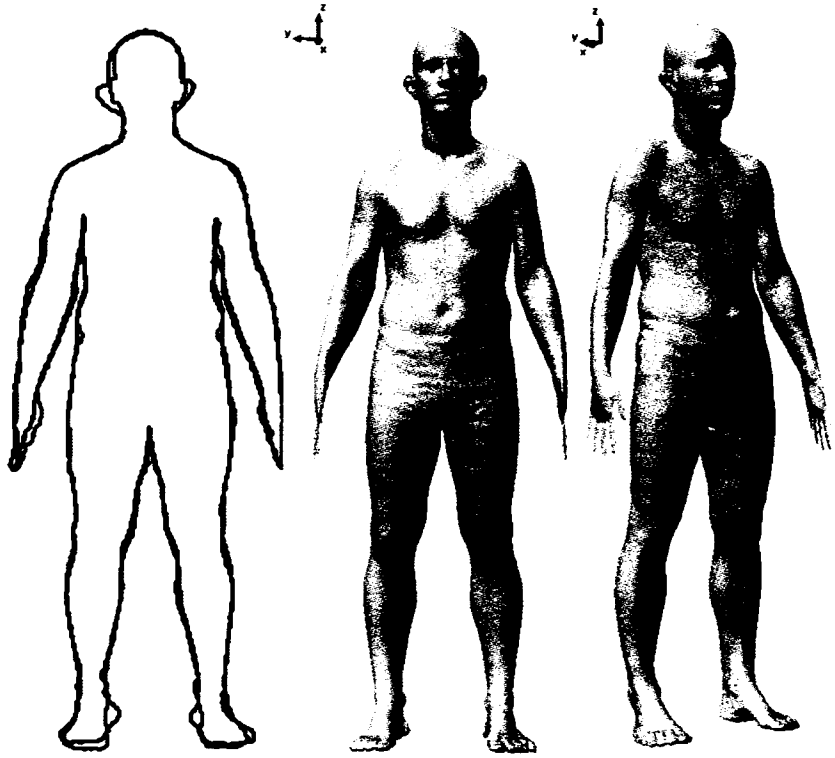


Figure 45: left: original contour (in green) and reconstructed 2D contour with 50 principal components (in red), middle: 3D reconstruction data with 50 principal components in 3D, right: diagonal view of the 3D reconstructed datum

We conduct some analysis on these worse results after increasing the number of 3D principal components. We conclude three sources of errors in this process: first, a reconstruction error in 2D if we select less number of principal components than all available; secondly, a mapping error from 2D to 3D by a linear relation; thirdly, a reconstruction error in 3D if less number of principal components are selected. There is no perfect mapping function, which would have an accurate mapping ability from 2D

coordinate to 3D coordinate. Therefore, there is always an error in each dimension of the mapped coefficient. As observed from experiments that, the first several principal components represent the most obvious variations, e.g. height, weight, and pose, while those less important components represent detailed information on the surface and they are more sensitive to the selection of weights compared to the first several components (because the reasonable interval for weight selection is narrower for less-influencing eigen-vectors). Therefore, given more components maintained, the shape of the result will get more affected by those less-influencing eigen-vectors. This has a more visible influence on the body surface: it becomes coarser and less human-alike.

Figure 47 to Figure 49 demonstrate another reconstruction example from an Asian male's contour. The reconstructed model becomes more bent after selecting more principal components in the 2D PCA-space.

We conclude that there is a hurdle on performance by reconstructing human bodies as a whole. This is even the case if the new subject differs from the database in pose or ethnicity.

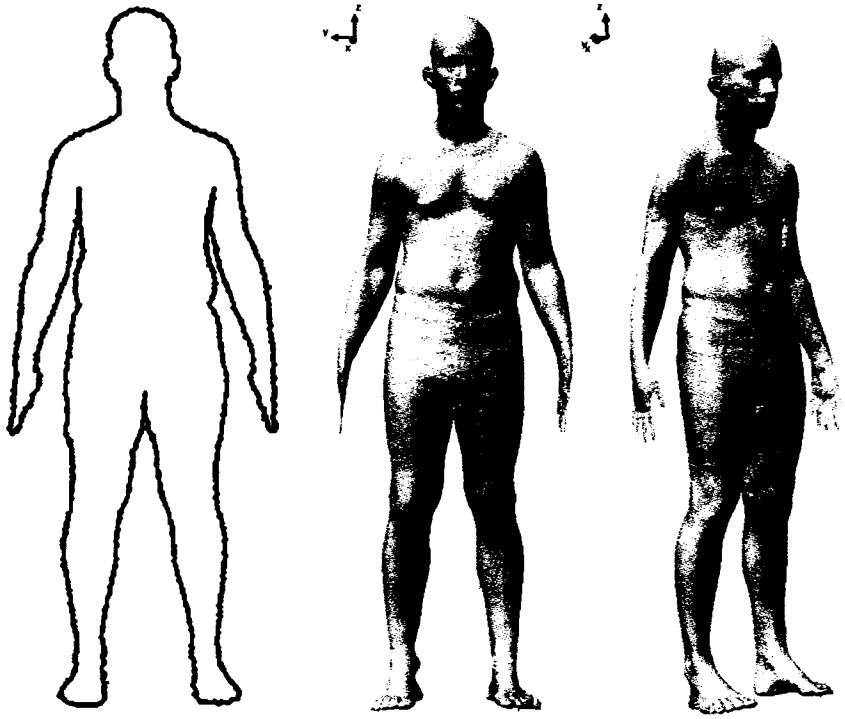


Figure 46: original contour (in green) and reconstructed 2D contour with all the 964 principal components (in red), middle: 3D reconstruction data with 50 principal components in 3D, right: diagonal view of the 3D reconstructed data

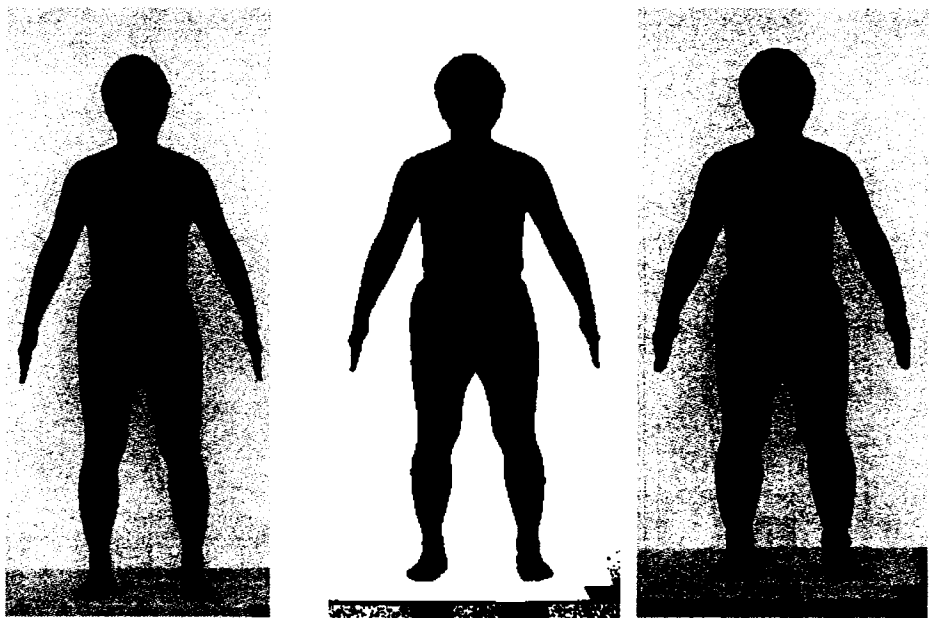


Figure 47: Contour finding for an Asian male subject (from left to right: the new image, its silhouette, and contour represented in green)

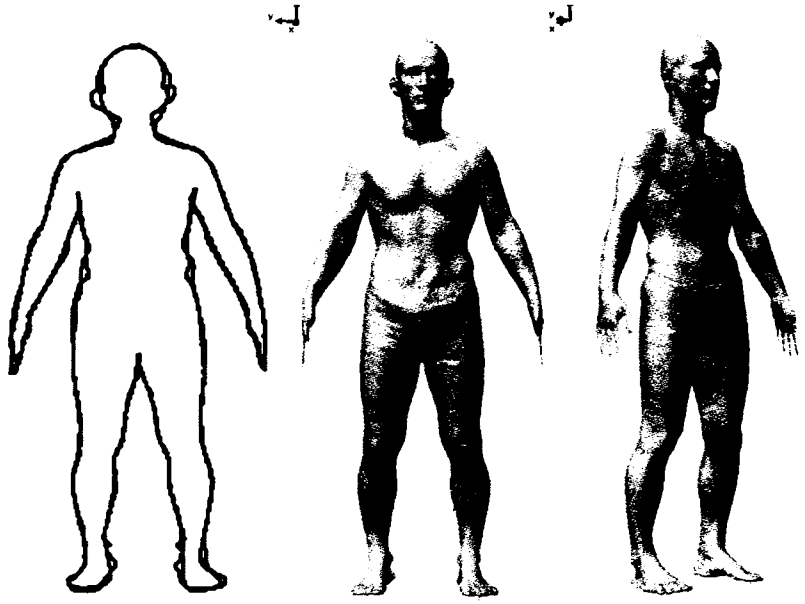


Figure 48: left: original contour (in green) and reconstructed 2D contour with 50 principal components (in red), middle: 3D reconstruction data with 50 principal components in 3D, right: diagonal view of the 3D reconstructed data

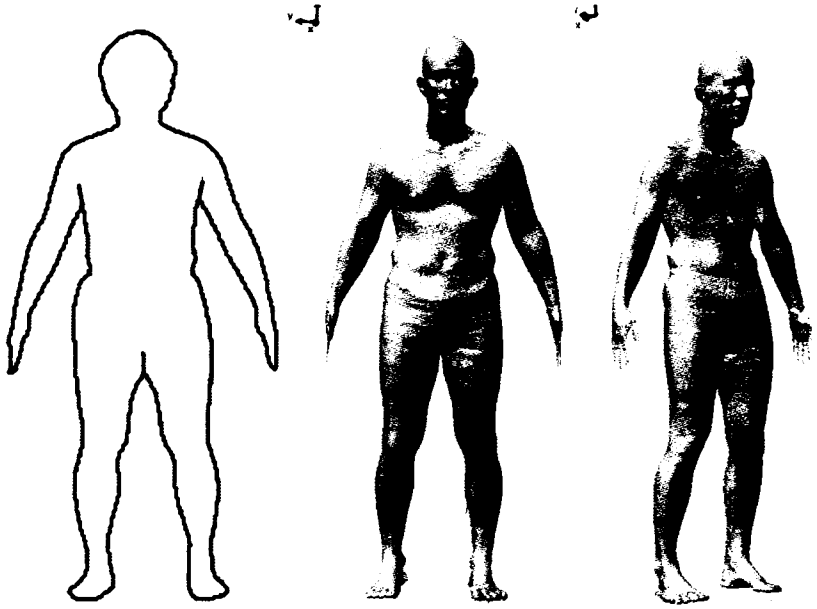


Figure 49: original contour (in green) and reconstructed 2D contour with all the 964 principal components (in red), middle: 3D reconstruction data with 50 principal components in 3D, right: diagonal view of the 3D reconstructed data

6.2 Reconstruction with segmented contours

We expect better reconstruction results from segmented human bodies. This is possible in that a richer expressiveness of human body segments is found in Chapter 4.

Before a reconstruction based on human segments is launched, we need first segment the new subjects' 2D contours in advance. This is not a problem in that we can always find a consistent representation according to feature points along those consistently represented whole-body contours.

Linear relationships between 2D and 3D PCA-spaces for each body part are built as similarly as those found from the whole body. After gaining an experience from the whole-body reconstruction, we select the number of 2D and 3D principal components all to be 40 in following experiments except introduced otherwise.

For the same subject used in Figure 44, here we conduct the segmented reconstruction as shown in Figure 50. We find an improvement by comparing the reconstructed results with those from the whole-body reconstruction. The final reconstructed body is more alike the target image in body ratio and local shape.

For the Asian subject's image previously used, we also perform a segmented reconstruction. The final results are shown in Figure 51. Here a more obvious improvement is observed compared to the whole-body reconstruction result.

We found in the whole-body reconstruction that if the arm pose is different from that in the database, there is an artefact on the hands; however, after we separately processing body parts, we find the reconstructed segments maintain good shapes.

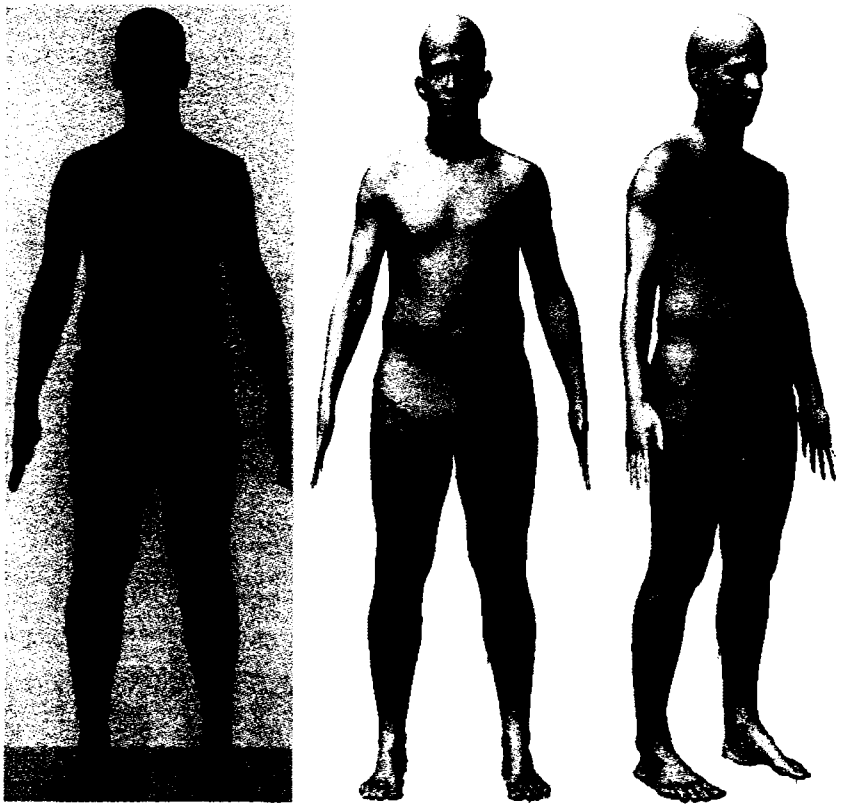


Figure 50: Segmented reconstruction of the subject (from left to right: the target image, front view of the reconstructed segments, and diagonal view of segments)

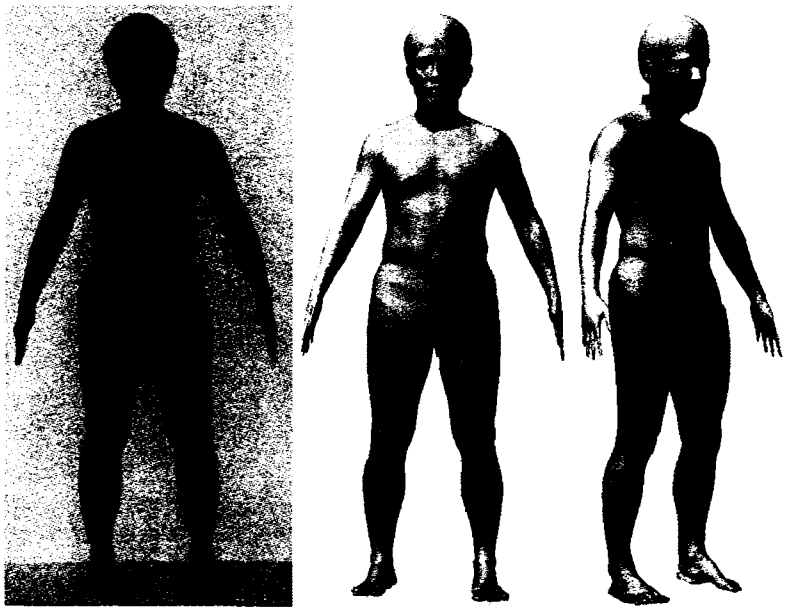


Figure 51: Segmented reconstruction result for the Asian subject

6.3 Stitching process for segmented reconstructions

After getting the reconstructed results based on segmentations, we need to merge them to build a complete model. Our idea is to keep the torso unchanged, and to stitch other neighbouring body parts onto the torso.

We start from building five RBF networks for the head, both arms, and legs separately. The RBF network is built upon corresponding vertex pairs including those on the overlapping patch between certain body part and the torso. To maintain local shape of each body part, we also select certain number of controlling points on those non-overlapping patches and set each vertex's target position as it is (for example, in stitching the head to the torso, we selected 80 vertices evenly from the head model, 22 of which are from the overlapping patch. The number 22 is selected in both maintaining the head shape and in making a good matching result). Figure 52 shows the stitched results.

We do find decreased distances between neighbouring segments after the stitching. For example, the crevice between the head and the torso in the Asian subject model was filled while the shape of the head is well maintained.

To make the stitched results smoother, we continue our stitching process by aligning another lower-resolution generic model of 100K triangles to these stitched results (see Figure 53 for the smoothed results). We used this lower-resolution model in expectation that the artefacts between neighbouring body parts would be removed after alignment. The alignment process is similar to that in the consistent parameterization process introduced in chapter 3. And the 100K generic model is a sub-sampled model from the previously used 328K generic model.

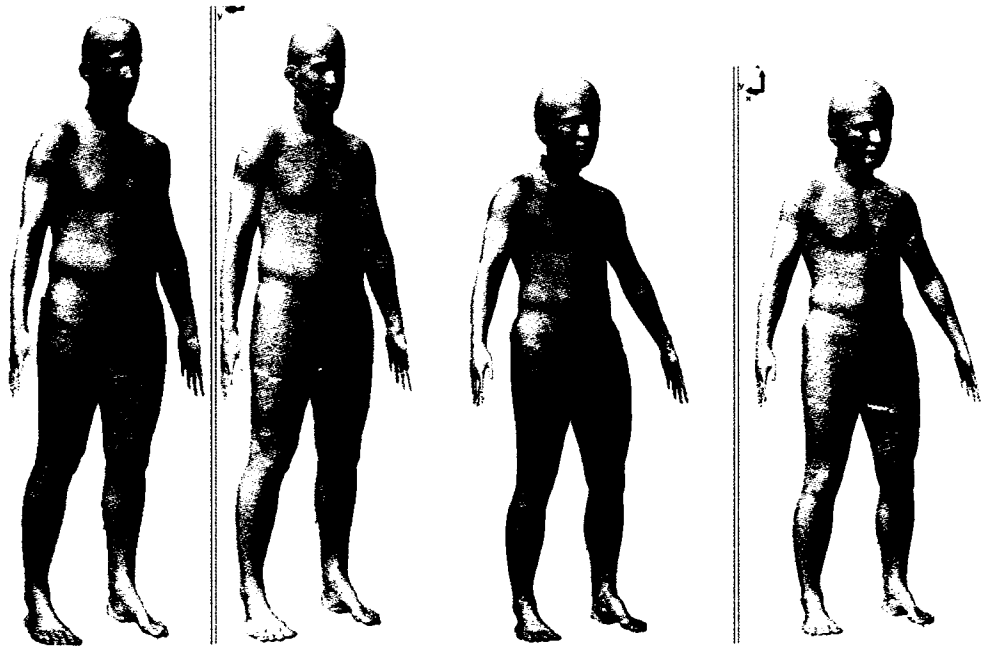


Figure 52: Stitching results with RBF (the colour models are segmented results, and the grey ones are corresponding stitched results)

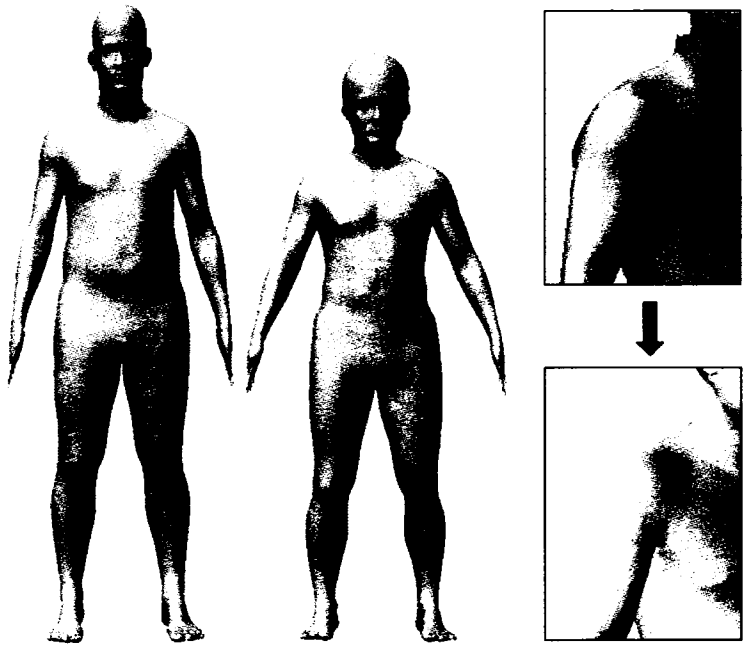


Figure 53: Further stitched results with a 100K lower-resolution model (the right snapshots on the shoulder and neck presents an obvious smoothing effect of our approach)

Stitching results shown in Figure 53 remove the artefacts being observable in Figure 52 while the surface details are partly lost because of the representation with this lower-resolution model.

6.4 Experiment on a toddler's photograph

In the previous section, we introduce the reconstructions of two subjects, one of Caucasian and another of Asian. The Caucasian person is not in the database but it belongs to the distribution of the database. The Asian subject is neither in the database nor among the distribution. For these subjects, we successfully reconstructed their shapes. Here we challenge the robustness of our methodology by reconstructing a 3D body from a toddler's front image.

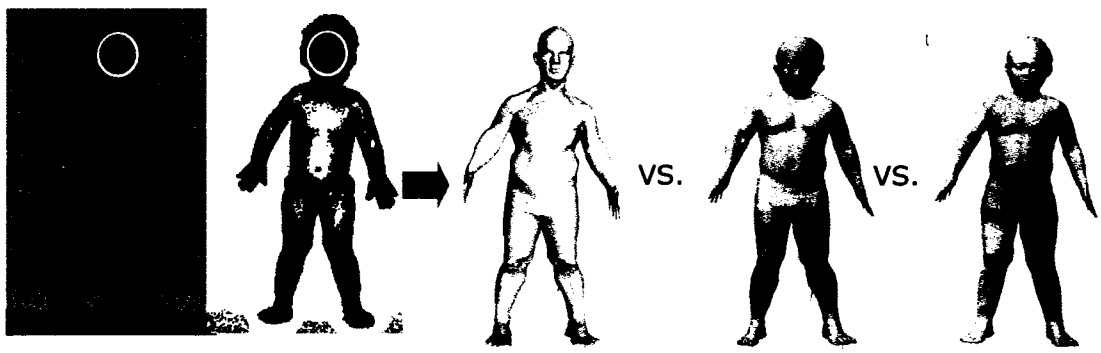


Figure 54: Segmented reconstruction results of a toddler's image (from left to right: the target image and the detected contour, whole-body reconstruction on a database of 964 male bodies, segmented body reconstruction with a body database of 964 male bodies, and a segmented reconstruction with a database of 1294 bodies, 95% of which are female bodies)

Figure 54 shows that, even if the target subject is totally outside of the distribution of the database, the segmented reconstruction is also able to create a good result. And the result is much better than that from whole-body reconstruction. We also observe that the reconstruction result from male database shows more muscles especially on lower legs

comparing the one reconstructed from the female database. And the shape on the waist is different because of the difference between male and female body characteristics.

Front and diagonal views of the resultant continuous meshes are placed together in Figure 55. The head shape of the toddler is unnatural due to her hairstyle. We are able to capture the approximated skull shape for an individual with either short hair or a ponytail as allowed in the CAESAR dataset; however, the toddler's hairstyle in the photograph is non-standard, resulting in the unnatural head shape.

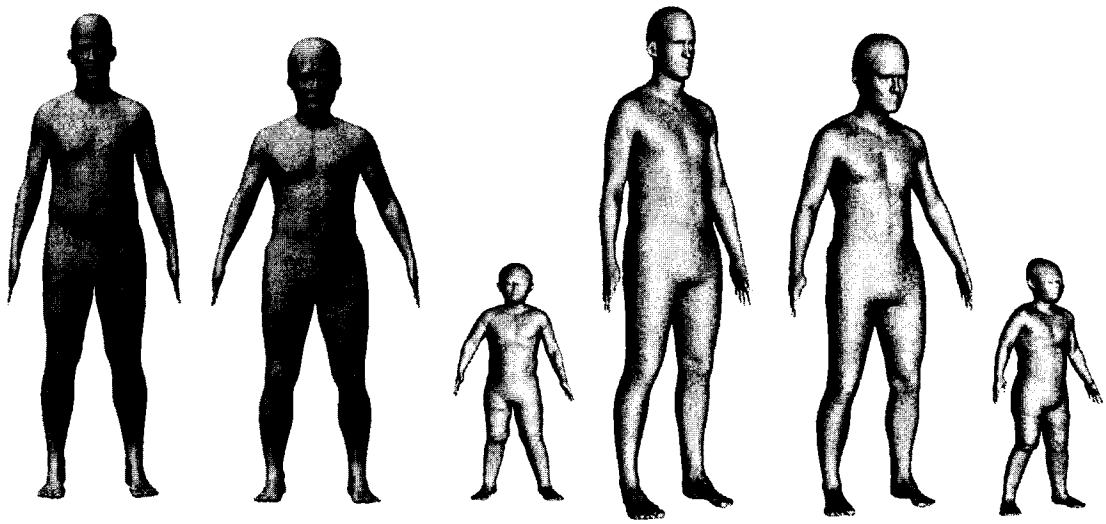


Figure 55: Final reconstruction results for the three subjects in their life-heights

6.5 Validation of results

We compare in Figure 56 the side-view of the reconstructed results with side-view photographs of the Caucasian and Asian subjects. The side view photographs are not used for 3D reconstructions.

Next we conduct a quantitative analysis on the accuracy of our segmented body reconstruction by selecting a 3D body from the CAESAR database. This 3D body is not

in the training database and we map it to 2D for getting its silhouette and therefore its contour using the same mapping settings as what were used in the training process.

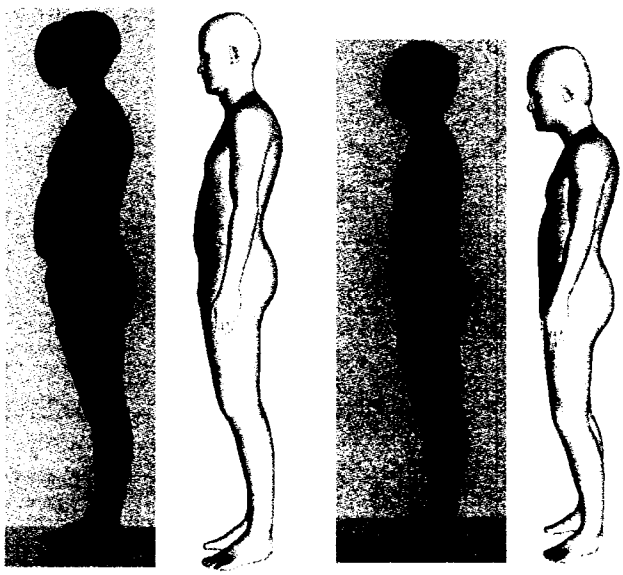


Figure 56: Comparisons between side-views of reconstructed 3D models with that of the original subjects.

We select a male body from the CAESAR and follow a similar process as what we did in reconstructing the Caucasian subject. The original data, reconstructed body and a comparison between these two models are shown in Figure 57.

Table 4 lists the statistical results after comparison. We compare the segmented reconstruction result (the data model) to the original model (the reference model). Therefore, the results are calculated by finding the nearest neighbour for each of the 50002 vertices in the reconstructed model. This comparison is automatically finished in the "IMInspect" component of PolyWorks.

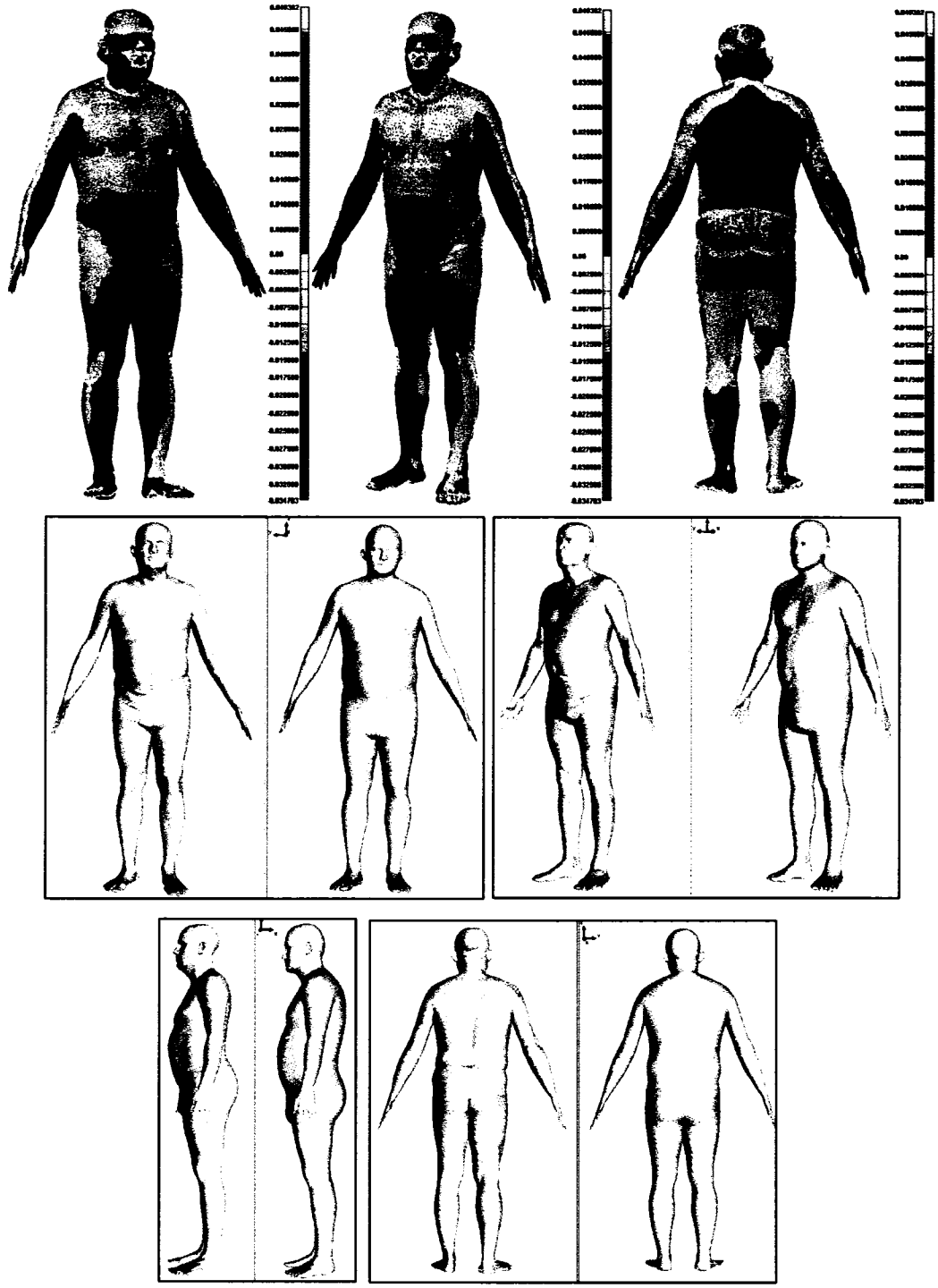


Figure 57: Visual verification results (upper row from left to right: difference by comparing the segmented reconstruction result to the original model; second row and third row: comparing the original model with the reconstructed model from different views)

Table 4: Summations of the error calculation between the segmented reconstruction result and the original model

Table Type	Data to Reference (Data Point)
Number of Points	50002
Mean	0.002552 m
Standard Deviation	0.011970 m
Root Mean Squared Error	0.012238 m
Max Error (+)	0.049382 m
Max Error (-)	-0.034703 m

For a new female subject "csr0379a" from the database, we conducted the segmented reconstruction on its contour. Figure 58 demonstrates comparisons between the original contour and reconstructed segment contours, and a comparison between the original contour and the mapped contour of the finally reconstructed 3D mesh. Although the previous comparison does not show obvious differences, there is a distance between the mapped contour of the segmented reconstruction mesh and the original contour. This distance results from the lost information after 2D PCA, the mapping error from 2D PCA space to 3D PCA space and the lost information after 3D PCA. There is only translation on the reconstructed contour from the original one, but the overall contour maintains very similar shape to that of the original contour.

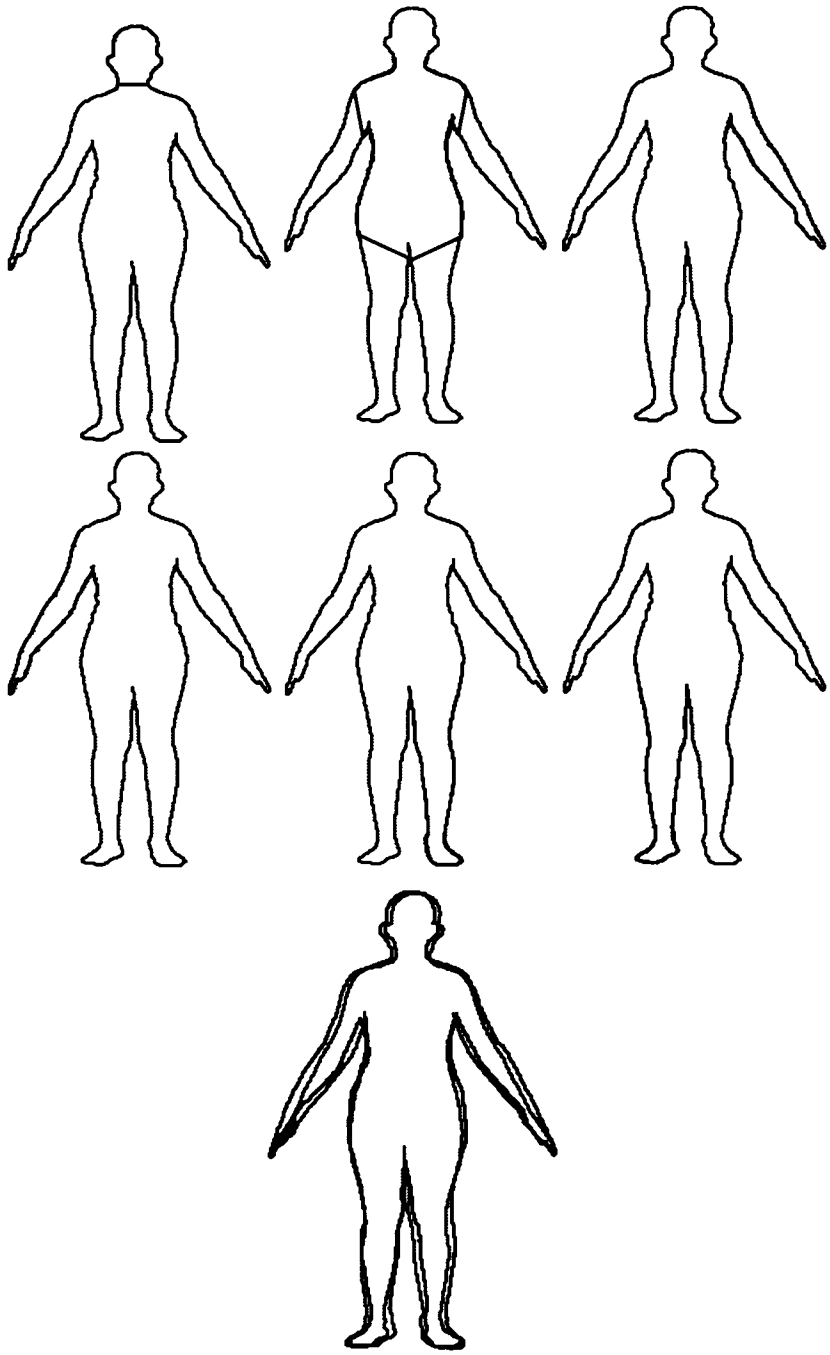


Figure 58: Comparisons between the reconstructed 2D body contours with the original contour (the first and second rows: reconstructed 2D segment contours in red with original contour in green; the third row: mapped contour from the segmented reconstruction mesh in red with original contour in green)

To make our validation more persuasive, we extract from the consistently represented database some data, map them to get their 2D silhouettes and conduct whole-body and segmented reconstructions respectively. By comparing the reconstructed results to the database, we get the comparison in Table 5. The testing examples are four Asian females (0249, 0270, 0256 and 0214), four Caucasian females (0401, 0136, 0158, and 0161), four African females (0380, 0213, 1258 and 0379), four Asian males (0253, 0209, 0169 and 0112), four Caucasian males (0234, 0144, 0146 and 0148) and four African males (0232, 0279, 0316 and 0441).

To get a more straightforward impression on the results from Table 5, we plot two figures on comparison between the whole-body and segmented reconstruction results. The numerical validation results in Figure 59 on the mean error and in Figure 60 on the standard deviation demonstrate that the segmented reconstruction has an overall higher accuracy than the whole-body reconstruction.

Data in Figure 61 and Figure 62 are all achieved with segmented reconstructions. We plot in Figure 61 the mean errors and standard deviations within the male and female groups. The lines demonstrate a better performance for reconstructing females. We also plot in Figure 62 the statistical results within the Asian, Caucasian and African groups. The mean errors for Asian subject are overall smaller than those in other groups. The Caucasian group has the most number of subjects whose standard deviation is smaller than that in other groups. We also find the standard deviations for African subjects have a smoother variation than that of other groups. Figure 63 to Figure 68 show all the visual comparisons between original scan, whole-body and segmented reconstructions.

Table 5: A detailed statistical results by comparing the whole-body and segmented reconstruction results with the original 3D data (Asian females: 0249, 0270, 0256 and 0214; Caucasian females: 0401, 0136, 0158 and 0161; African females: 0380, 0213, 0258 and 0379; Asian males: 0253, 0209, 0169 and 0112; Caucasian males: 0234, 0144, 0146 and 0148; African males: 0232, 0279, 0316 and 0441)

Whole-body Reconstruction						Segmented Reconstruction				
Sub#	Mean	StdDev	RMS Err.	MaxErr +	MaxErr -	Mean	StdDev	RMS Err.	MaxErr +	MaxErr -
0249	0.015672	0.026346	0.030655	0.115788	-0.07289	0.004982	0.035182	0.035533	0.100307	-0.10197
0270	0.007997	0.018166	0.019848	0.073651	-0.04645	-0.00086	0.027668	0.027681	0.083965	-0.08698
0256	0.000159	0.020484	0.020485	0.070531	-0.07226	-0.00152	0.011739	0.011837	0.032237	-0.04611
0214	0.004753	0.015706	0.01641	0.050032	-0.0498	0.000093	0.018451	0.018451	0.054621	-0.06576
0401	0.011861	0.049656	0.051053	0.151743	-0.15033	0.043436	0.05882	0.07312	0.249672	-0.13673
0136	0.005802	0.016678	0.017659	0.067133	-0.07153	-0.00053	0.015498	0.015507	0.065301	-0.05875
0158	0.000398	0.01963	0.019634	0.06345	-0.0752	-0.00173	0.013817	0.013924	0.050595	-0.04103
0161	0.003152	0.016003	0.01631	0.060226	-0.07664	-0.00269	0.0203	0.020478	0.050617	-0.06847
0380	0.010059	0.024986	0.026935	0.10162	-0.08316	0.000242	0.017804	0.017806	0.064795	-0.05306
0213	-0.00161	0.019803	0.019868	0.080492	-0.05855	0.001569	0.016008	0.016085	0.070931	-0.07177
0258	0.002177	0.016326	0.01647	0.068876	-0.0625	0.003653	0.020214	0.020542	0.085647	-0.06545
0379	0.0104	0.021213	0.023626	0.074388	-0.05075	0.002523	0.028788	0.028899	0.074113	-0.07855
0253	0.038425	0.045849	0.059821	0.211439	-0.09025	0.005013	0.043407	0.043695	0.126026	-0.12931
0209	0.018959	0.03493	0.039743	0.155088	-0.07906	0.003437	0.028943	0.029147	0.090669	-0.09195
0169	0.004975	0.01571	0.016479	0.111276	-0.05205	-0.0028	0.01639	0.016628	0.07403	-0.06663
0112	0.015776	0.033385	0.036925	0.228306	-0.07685	0.004079	0.027338	0.027641	0.076325	-0.08786
0234	0.02492	0.036347	0.044069	0.179319	-0.07419	0.008264	0.040374	0.041211	0.105287	-0.11148
0144	0.005368	0.024024	0.024616	0.12526	-0.06615	0.005902	0.020878	0.021696	0.116694	-0.09211
0146	0.003151	0.018378	0.018647	0.102489	-0.08933	0.003747	0.014827	0.015293	0.074498	-0.04253
0148	0.00883	0.020379	0.02221	0.194068	-0.06772	0.0032	0.019536	0.019797	0.062894	-0.07018
0232	0.012241	0.023505	0.026501	0.106148	-0.05219	0.003305	0.021127	0.021384	0.072593	-0.07239
0279	0.011812	0.02603	0.028585	0.23133	-0.06228	0.005416	0.025697	0.026262	0.121699	-0.12166
0316	0.005147	0.017149	0.017905	0.193196	-0.05163	0.0021	0.018635	0.018753	0.075877	-0.0461
0441	0.006446	0.023597	0.024462	0.219901	-0.08126	0.004944	0.020816	0.021395	0.096772	-0.07882

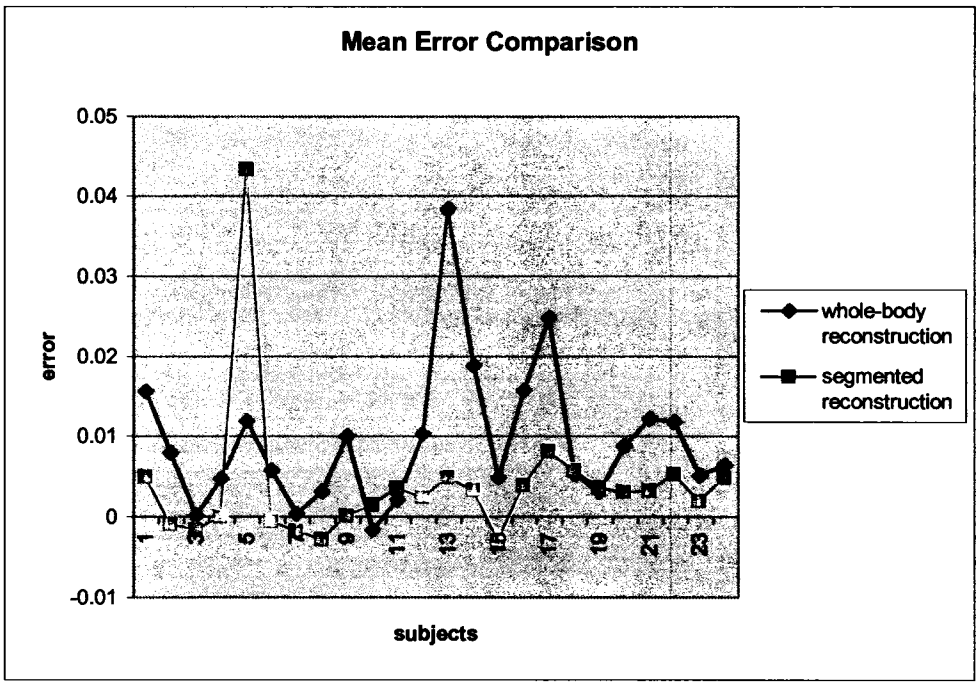


Figure 59: Mean error comparison between the whole-body and segmented reconstructions for all the 24 subjects

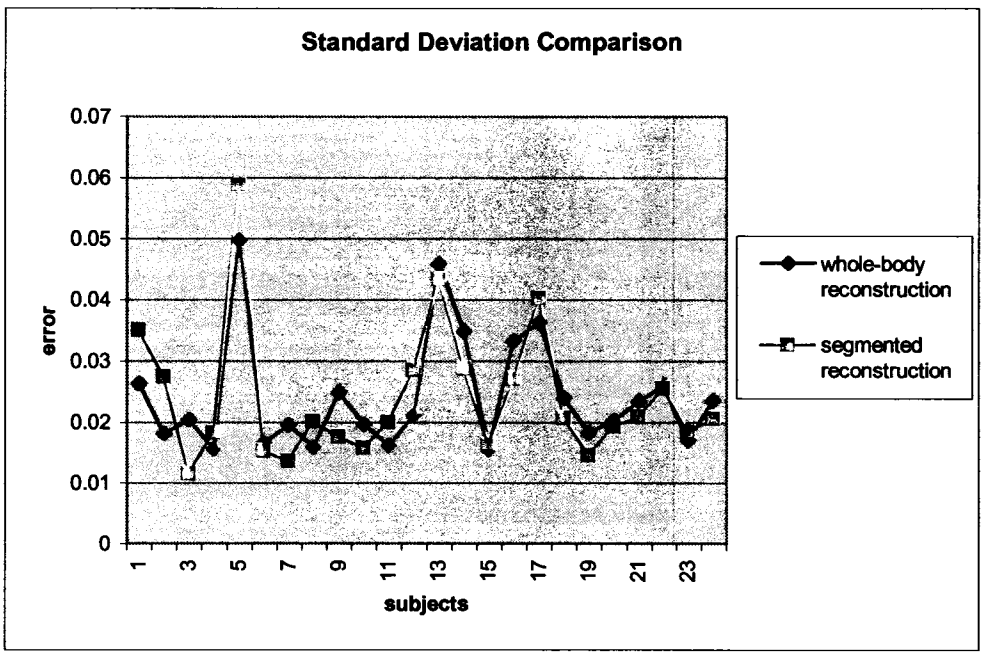
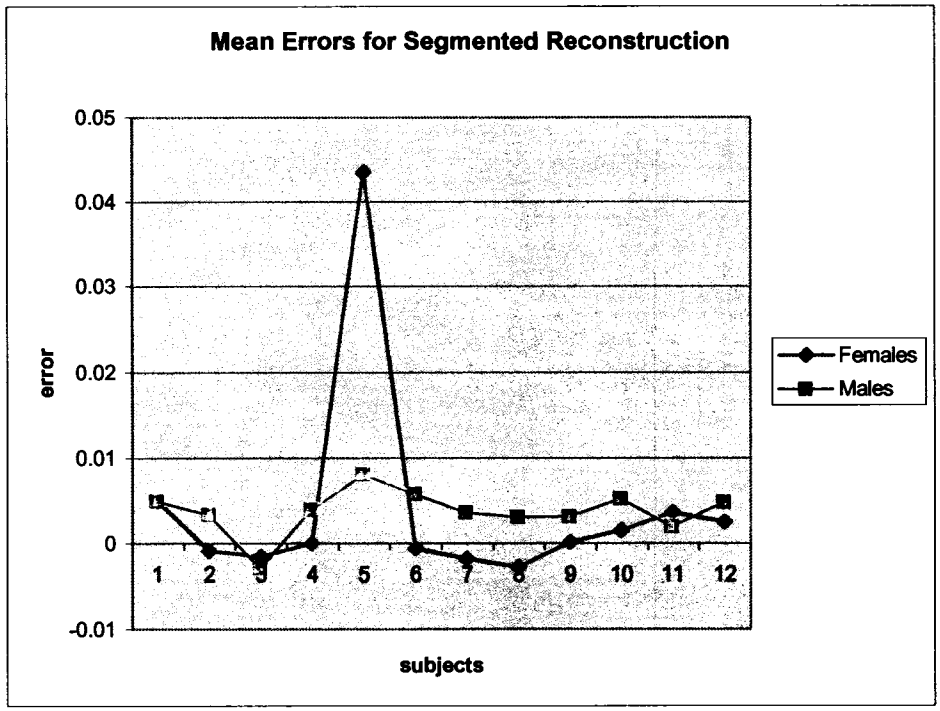


Figure 60: Standard deviation comparison between the two approaches for all the 24 subjects



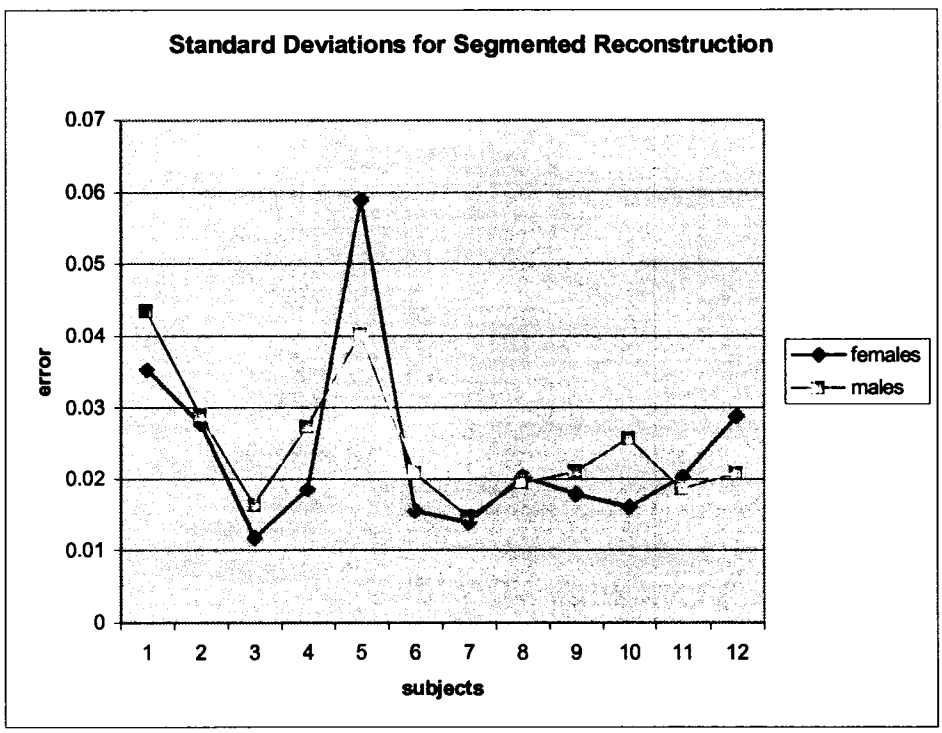
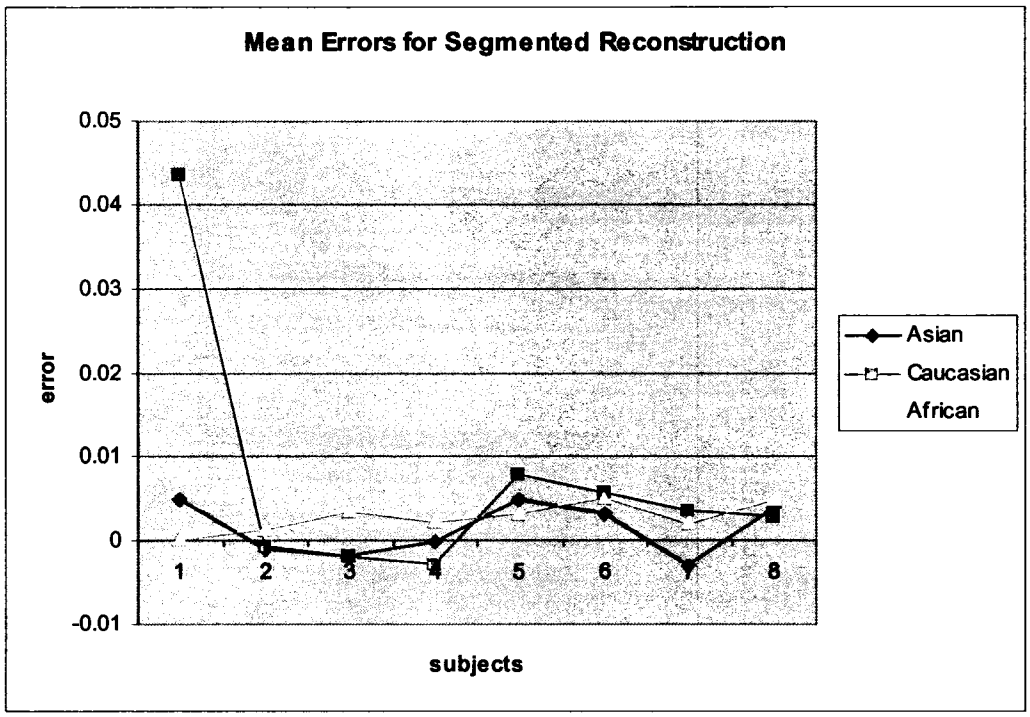


Figure 61: Comparisons between Male and Female groups with the segmented reconstruction (overall mean errors and standard deviations are smaller for females)



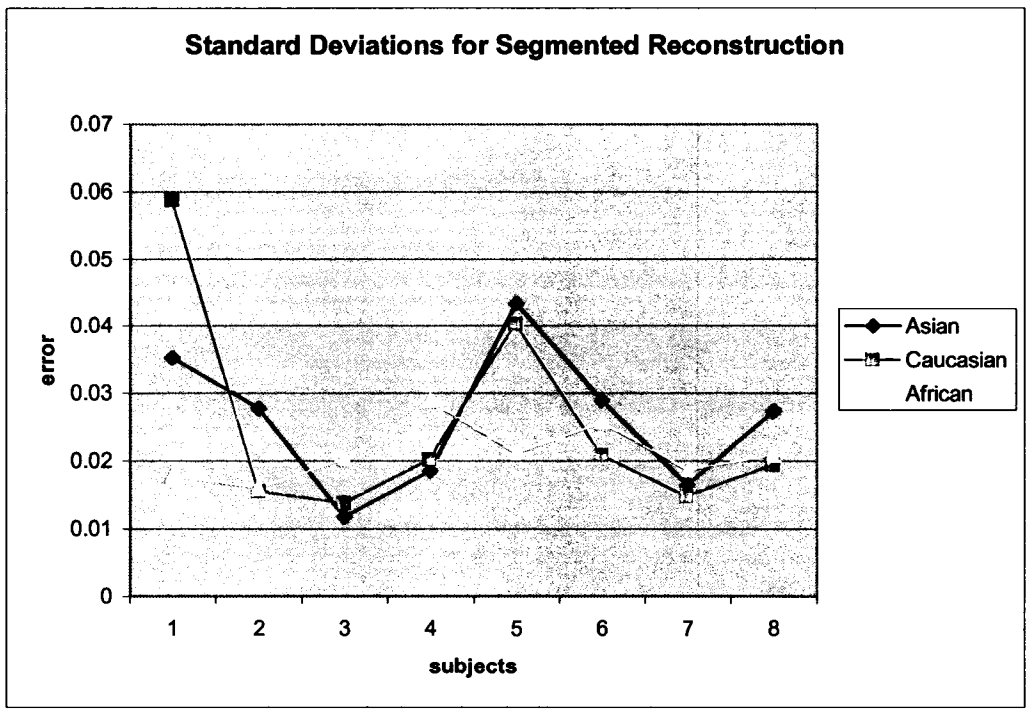


Figure 62: Comparisons between ethnic groups with the segmented reconstruction (overall the mean error for Asian subjects are smaller than that for other groups, the standard deviation for Caucasian subjects are smaller than that for other groups)

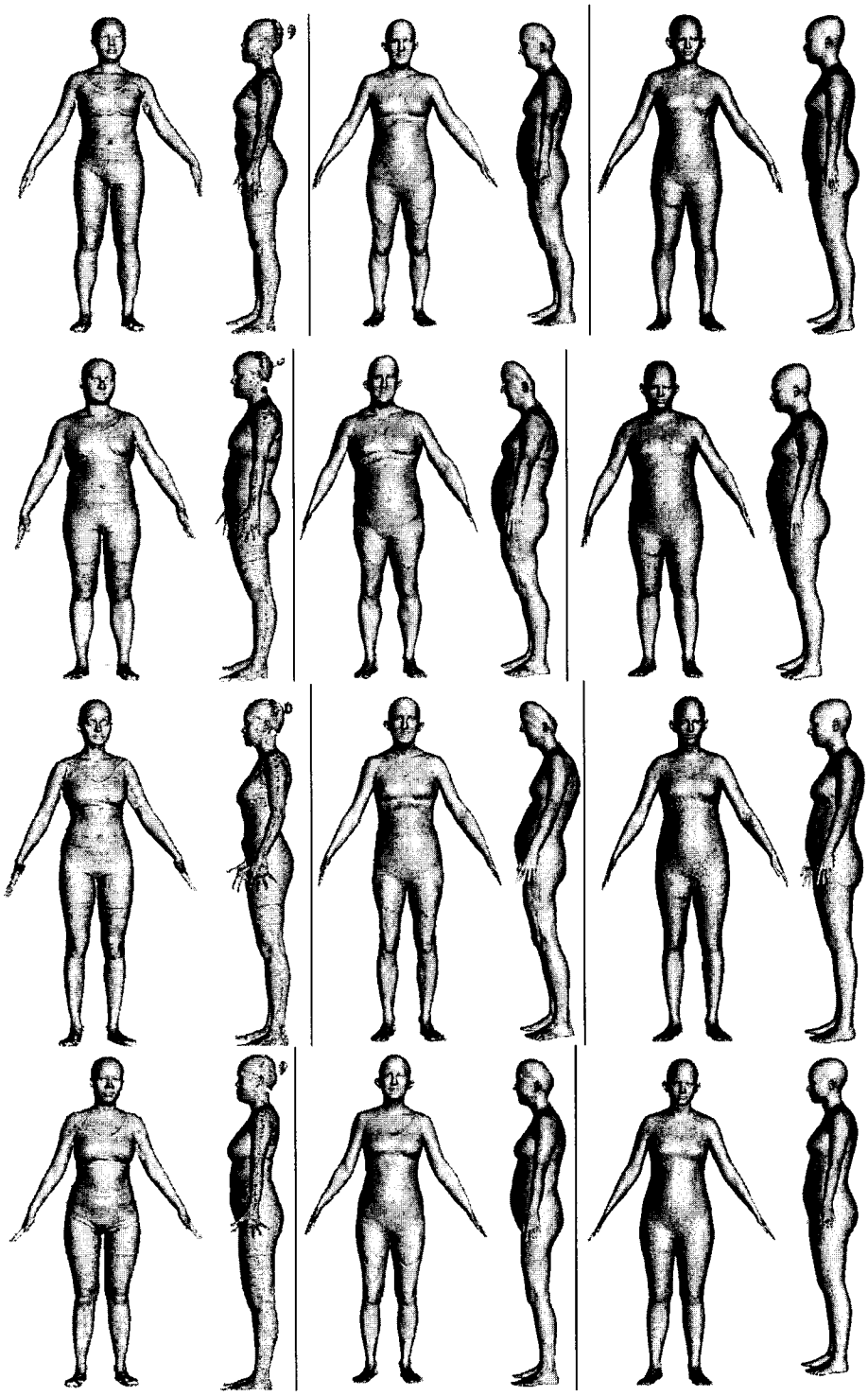


Figure 63: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0249, 0270, 0256 and 0214)

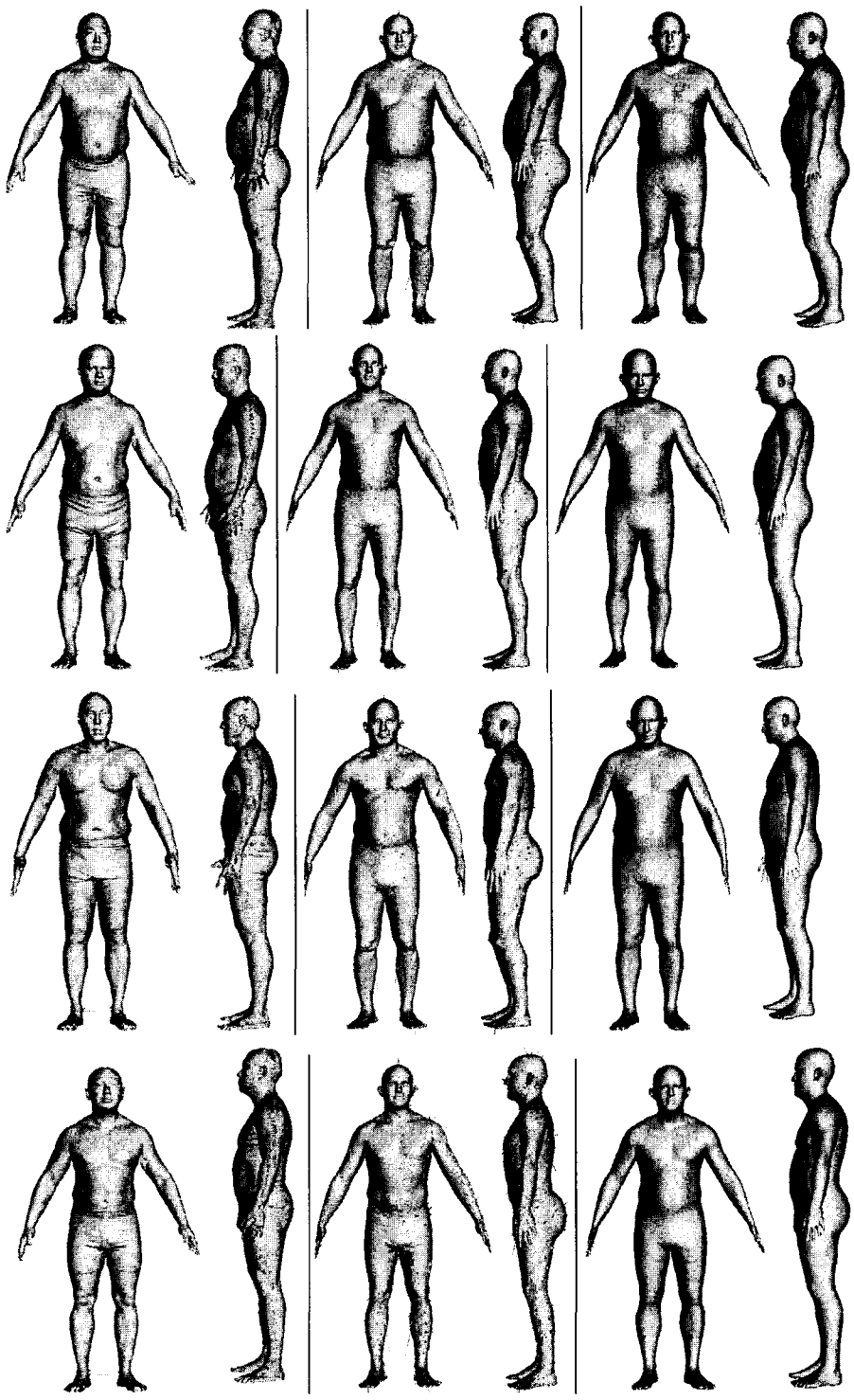


Figure 64: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0253, 0209, 0169 and 0112)

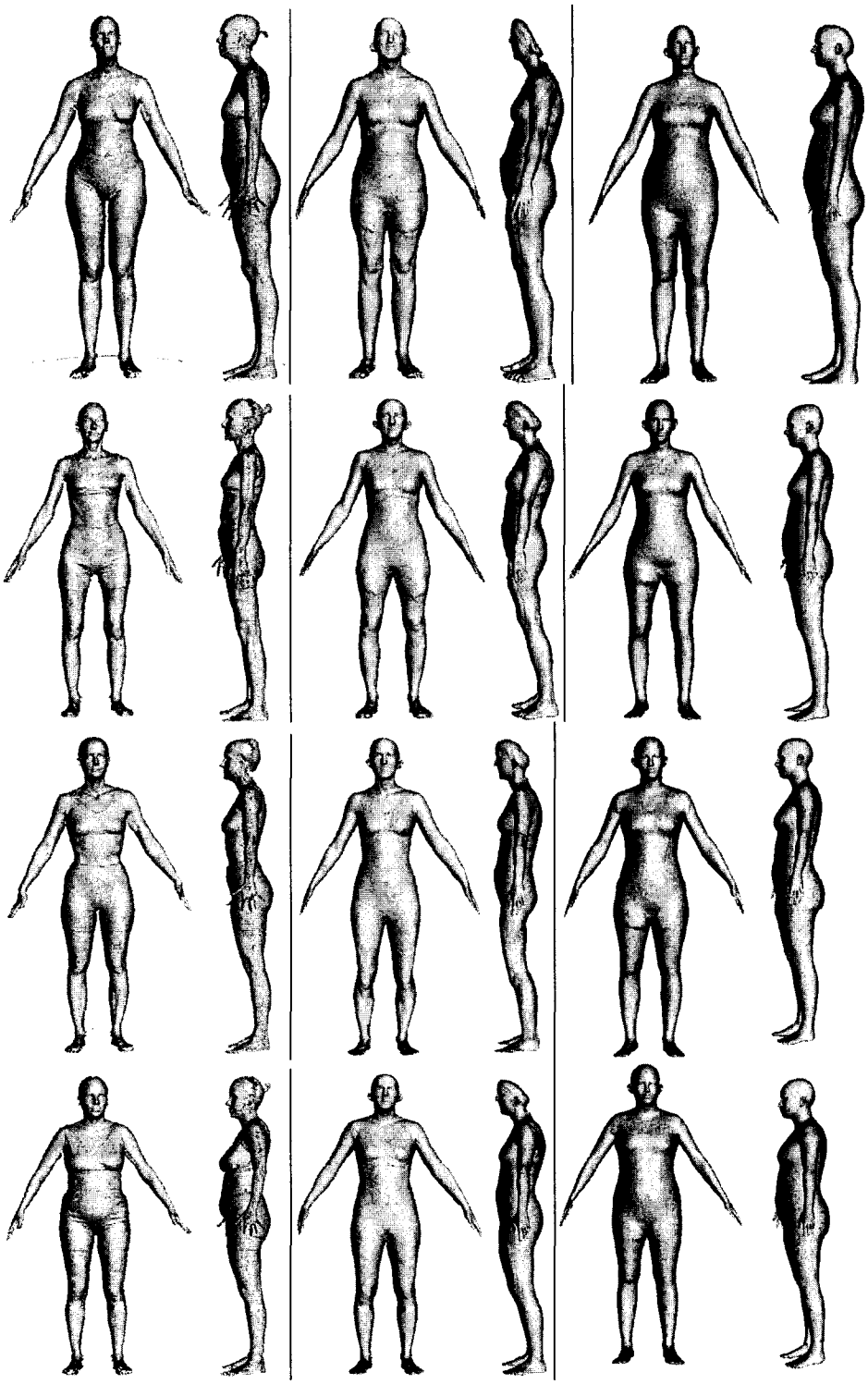


Figure 65: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0401, 0136, 0158 and 0161)

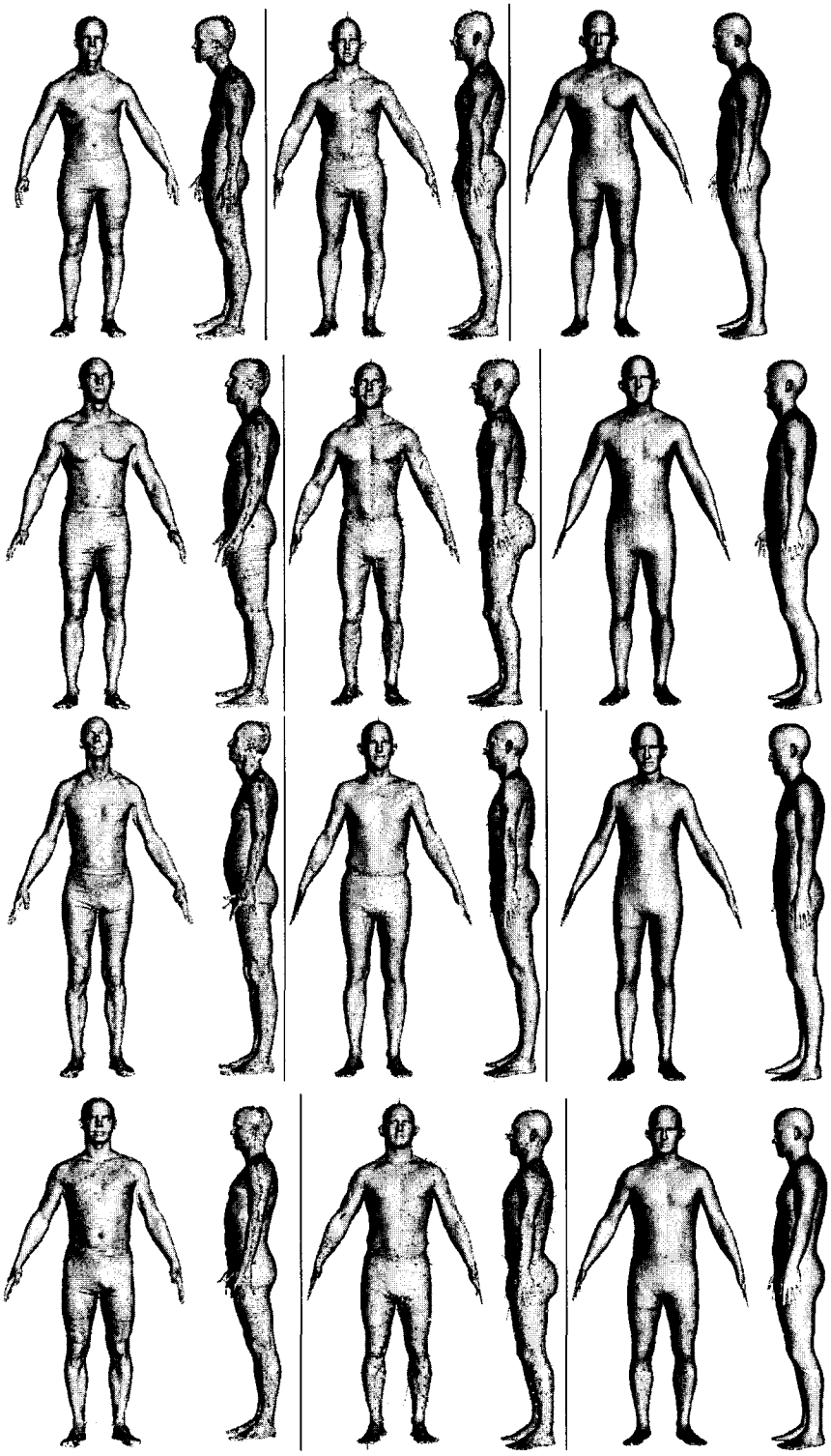


Figure 66: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0234, 0144, 0146 and 0148)

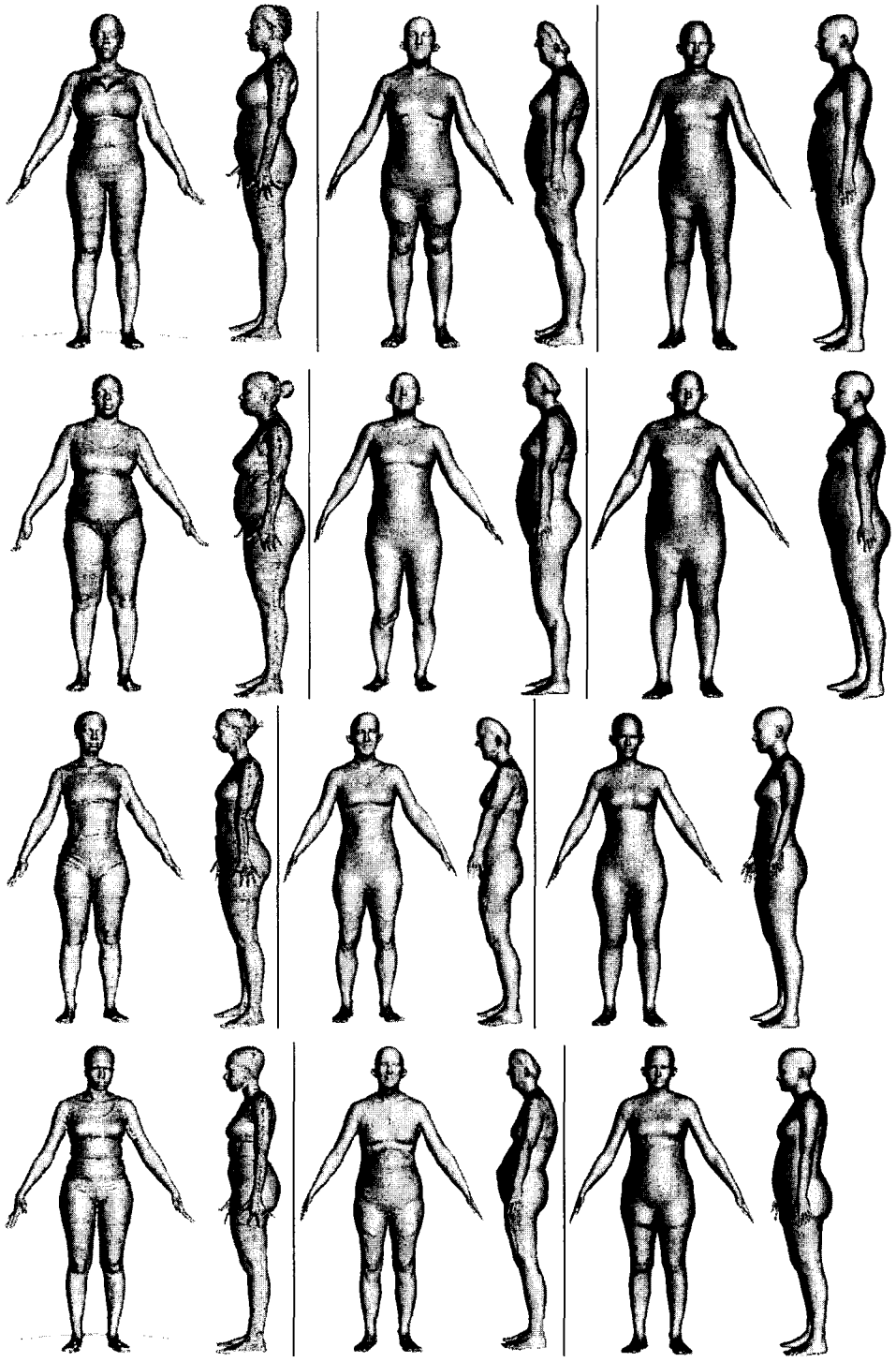


Figure 67: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0380, 0213, 0258 and 0379)

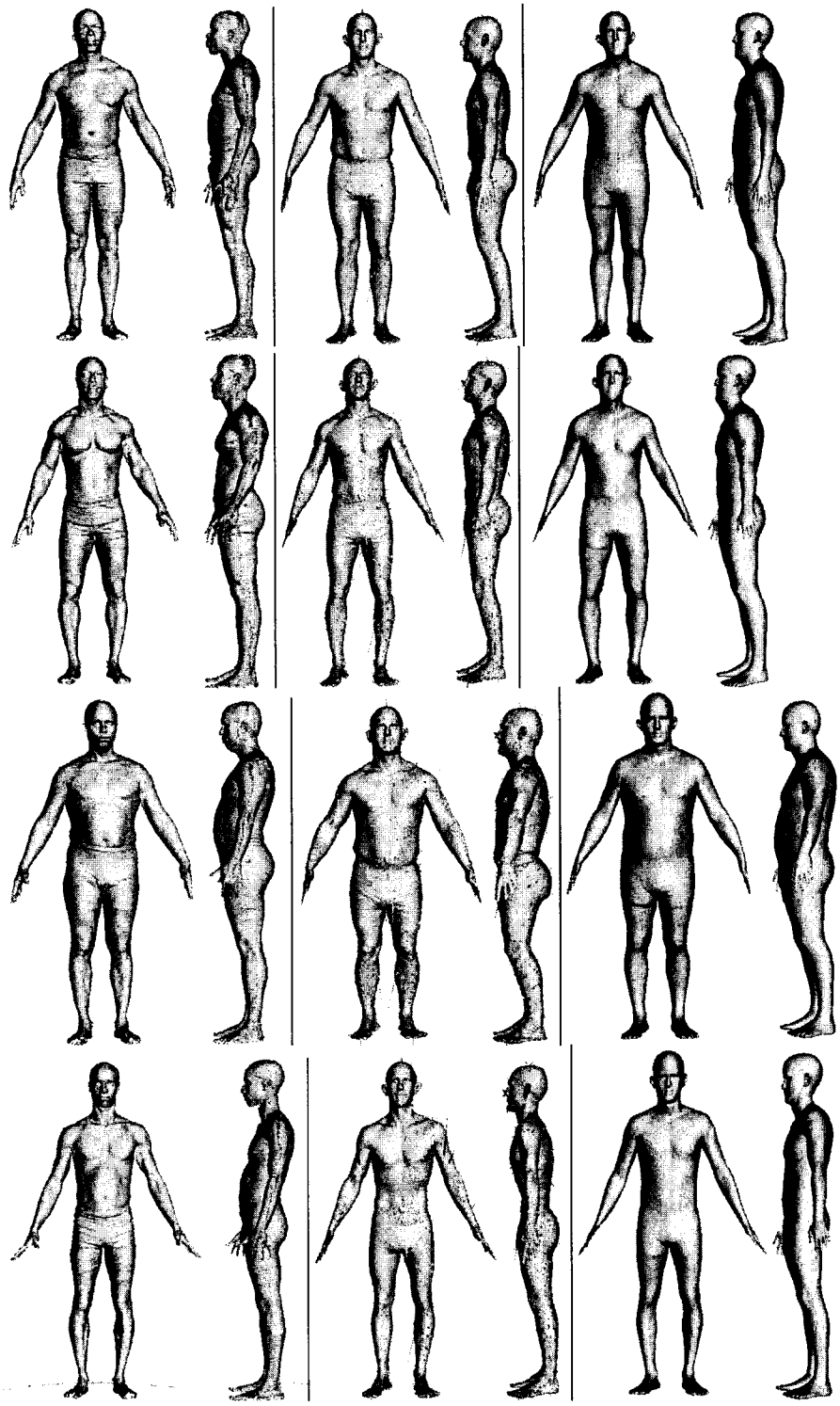


Figure 68: Comparisons between the original scan, whole-body and the segmented reconstruction results (from the first row to the fourth row: 0232, 0279, 0316 and 0441)

6.6 Texture mapping

After building a 3D model, we are thus able to get its texture by projecting input photograph onto the model. We use front and back-view photographs for acquiring texture of the Caucasian and Asian subjects as shown in Figure 69. While for the toddler, we had a difficulty in taking a back-view photograph, so we only map the front texture.

Texture image is built by putting the front and back view images into a single image. Therefore, in this texture image, the left half is the front image of certain subject, while the right half is the picture from a back view. The texture image for the toddler only has a front-view image.

We applied similar texture mapping approach as used in [27][35]. Projection creates texture coordinates automatically where the result is shown in Figure 69.

For each vertex on the 3D model, we need first know which vertices will be textured with the front-view image and which other vertices will be textured from the back-view image. Thus we make another partition on the generic model with 100K triangles by selecting the front half of the whole body and find all the vertices along its boundary. We need to solve a problem along the boundary before the mapped texture becomes continuous.

The textured VRML file of the result 3D model consists of four parts: 3D data points, 3D triangle indices, 2D data points, and 2D triangle indices. In constructing this textured model, we first copy from the reconstructed 3D model its 3D data point and triangle index information into the first two parts of the textured new model. For each vertex on the 3D model, we map it to the front-view texture image if this vertex lies on the front

patch. Accordingly, if it lies on the back patch, we map it to the back-view image. If this vertex is along the boundary of the front patch, we need map it both to the front and to the back view images. Therefore, for the third part of the textured 3D model, i.e. the 2D data points, the length of all the 2D coordinates will be longer than that of the first part, i.e. the 3D data points. The difference of these two lengths is the number of vertices along the boundary, which are mapped twice and the mapped coordinates onto the back-view image are appended to the 2D data points queue.

Our next step is to copy the 3D triangle indices into 2D triangle indices and update them. By visiting each triangle, we first decide whether any of the three vertices are from the back patch. If so, we will map all the three vertices to the back-view texture image; otherwise, all three vertices will all be mapped to the front-view texture image. If the vertex is determined to be mapped to the back-view image, we need to update its index in this triangle; otherwise, we keep its index as it is.

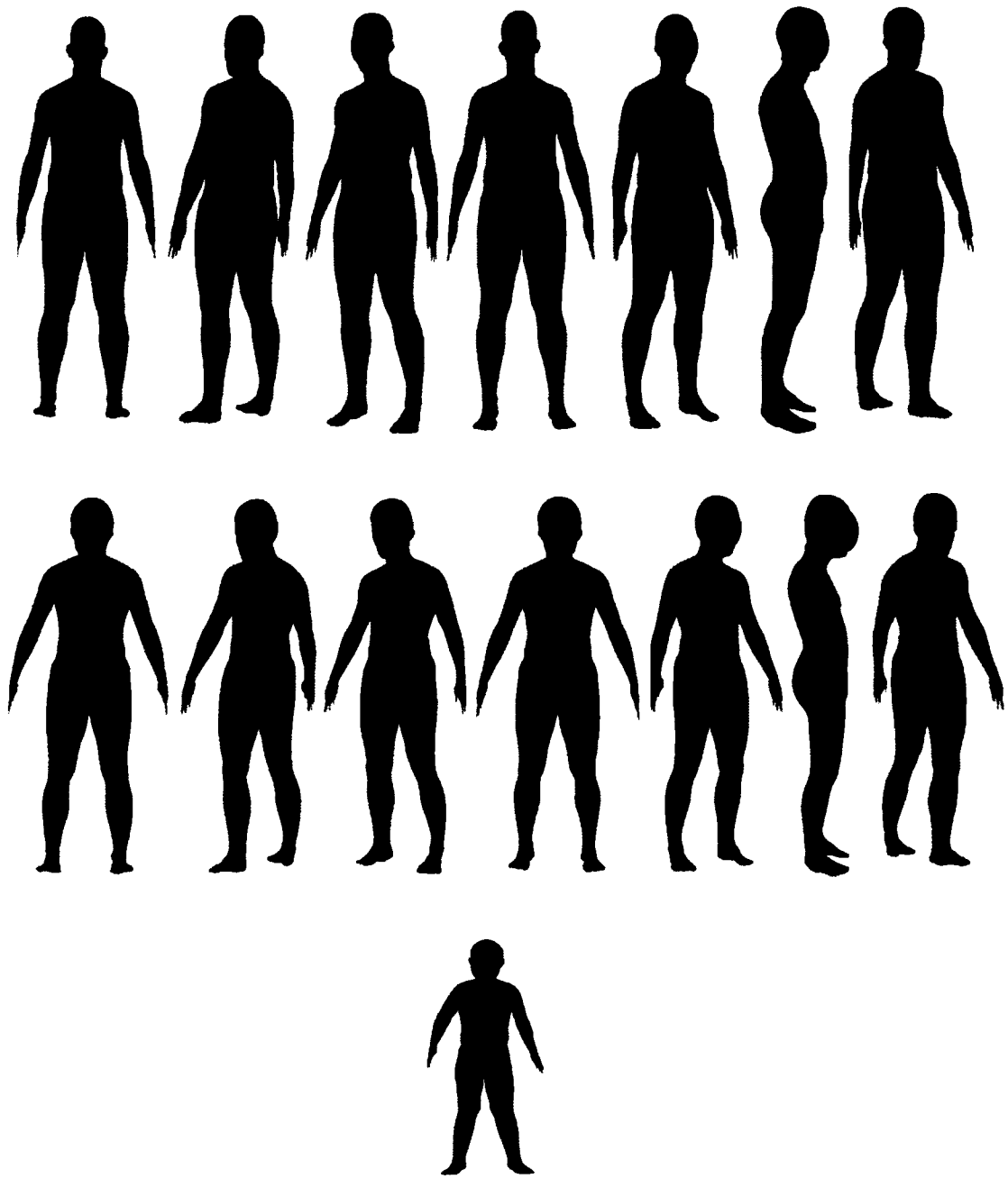


Figure 69: 3D reconstruction results after texture mapping

Chapter 7: Discussions and Conclusions

The market for human body modeling is expanding rapidly. Applications in mechanical engineering, medical appliances, and cloth design are all driving the research of human body modeling. Statistical approach for representing human body space has proved to be a good way in reconstructing new data, which is outside of the training database. The possibility of reconstructions of 3D data out of 2D information greatly simplifies the requirements and thus drives the application of human body modeling. Therefore, the following findings and conclusions may be of help for designers, researchers and engineers. [20]

7.1 Statement of results

Our contribution lies in two aspects, i.e. an efficient alignment process and a segmented body reconstruction approach meeting on-site application requirements. These are introduced in great detail as the main content of this thesis: the consistent parameterization and the 3D reconstruction out of 2D images.

We start from a high-resolution generic model, align it to raw scanned data by rough alignment with RBF network and fine alignment with a minimisation of defined errors (this two-step process is twice faster than Allen et al. [4]); therefore, we build a morphable 3D model with the principal components analysis. We also conduct a

consistent representation on contours of all the mapped silhouettes from 3D data, and then build morphable 2D models. Imbibing an idea from [3] on face modelling, we segment the generic body model into six parts, and similarly segment the contours into corresponding parts referring to some mapped landmarks.

Our assumptions that, coordinate vectors in 3D and 2D PCA spaces have a linear relationship help reconstruct a new 3D data in a very efficient way. We conclude the overall timing results in Table 6, where we recorded the time needed for reconstructions based on a database with 200 subjects. Overall the 3D processing costs longer time than what is needed in 2D. In 3D, most time is spent on conducting PCAs and project for getting weights, while getting weights for whole body costs the longest time. The reason for this lies in multiple reading of files from hard disk and in conducting multiple matrix calculations. Mapping for getting weights means to time the eigenvectors with each training mesh; in this case, we have 200×200 vector multiplications where each vector has a size of 492384. As all the 3D calculations were performed in *C* programming language, the matrix calculations costs a lot of time, while there are good timings for 2D calculations in Matlab, which is an optimised software for matrix calculations. Fortunately most of the 3D processing belongs to off-line training, while the off-line reconstruction of 3D meshes from segments can be finished in minutes. Experiments in 3D were conducted with C programming language on a PC of AMD Athlon 64×2 dual core processor at 2,200MHz and a memory of 4GB in size. Other experiments in 2D were conducted in Matlab language on a PC of Pentium 4 processor at 2.0GHz and a memory of 1GB in size.

Table 6: The overall timing (minutes) for each step of our approach

Space	whole-body PCA	project for weights	segmented PCA for head	segmented PCA for torso	segmented PCA for left right arm	segmented PCA for left/right leg	project for weights: head	project for weights: torso	project for weights: left/right arm	project for weights: left/right leg
3D	25	84.5	6	9	6	7	12	32	15	18
2D	2	0.5	0.1	0.2	0.1	0.1	0.1	0.2	0.1	0.1

	training for getting a relation	segmented reconstruction	reconstruction for a whole body	stitching the reconstructed segments	representing contour of a new picture
whole-body	1	N/A	0.78	4.5	0.5
head	0.1	0.1			
torso	0.1	0.2			
left arm	0.1	0.1			
right arm	0.1	0.1			
left leg	0.1	0.1			
right leg	0.1	0.1			

We have described how to construct 3D body shape with a single photograph, focusing on how to cover bigger range of poses, ethnic and age groups of input human subjects. Human subjects who do not belong to the ethnic or age group of the database are successfully reconstructed. Our Sub-divided PCA space based on segmented body database proves high potential to be more accurate than the whole-body reconstruction, and it is more flexible in data-driven modeling and animation.

7.2 Problems in current project and future research implications

The RBF helps speed up the alignment process, while the deformation on both arms tends to be affected by landmarks from other body parts. We are thinking about building different RBFs for each human body segment. Since RBF is more suitable for volumetric deformations, the separate processing may create better results. We also tried to replace current distance definition from Euclidian to geodesic one but we did not get better result.

Our present work only analyzes the contours of mapped silhouettes, so we have a plan in representing inner features, like the eyes, nose, mouth, navel locations. This more detailed representation along with segmented reconstruction will be able to create better reconstruction results. Considering traditional approach in aligning textures [3], this improvement could make up for the disadvantage of our methodology.

CAESAR dataset includes three scans per person in a standing pose, full-coverage pose and relaxed seating pose and the other two poses not-covered in this paper are also our future research topic.

References

[1] Vetter T, Poggio T. Linear object classes and image synthesis from a single example image. In: IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997;19(7):733-742.

[2] DeCarlo D, Metaxas D, Stone M. An anthropometric face model using variational techniques. In: ACM SIGGRAPH Computer Graphics Proceedings. 1998. p. 67-74.

[3] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. In: ACM SIGGRAPH Computer Graphics Proceedings. 1999. p. 187-94.

[4] Allen B, Curless B, Popovic Z. The space of human body shapes: reconstruction and parameterization from range scans. ACM Transactions on Graphics (TOG). 2003;22(3):587-594.

[5] Praun E, Sweldens W, Schroder P. Consistent mesh parameterizations. In: ACM SIGGRAPH Computer Graphics Proceedings. 2001. p. 179-84.

[6] Davis J, Marschner SR, Garr M, Levoy M. Filling holes in complex surfaces using volumetric diffusion. In: Proceedings of the First International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT). June 2002.

[7] Carr JC, Beatson RK, Cherrie JB, Mitchell TJ, Fright WR, McCallum BC, Evans TR. Reconstruction and representation of 3D objects with radial basis functions. In: ACM SIGGRAPH Computer Graphics Proceedings. 2001;67-76.

[8] Whitaker R. A level-set approach to 3-D reconstruction from range data. International Journal of Computer Vision. 1998;29(3):203-231.

[9] Allen B, Curless B, Popovic Z. Articulated body deformation from range scan data. ACM Transactions on Graphics (TOG). 2002;21(3):612-619.

[10] Hilton A, Starck J, Collins G. From 3D shape capture to animated models. In: Proceedings of the First International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT 2002). p. 246-57.

[11] Kahler K, Haber J, Yamauchi H, Seidel HP. Head shop: Generating animated head models with anatomical structure. In: Proceedings of the ACM SIGGRAPH Symposium on Computer Animation. 2002:55-64.

[12] Anguelov D, Srinivasan P, Koller D, Thrun S, Rodgers J, Davis J. SCAPE: Shape completion and animation of people. ACM Transactions on Graphics (TOG). 2005;24(3):408-16.

[13] Noh JY, Neumann U. Expression cloning, In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. 2001. p. 277-88.

[14] Seo H, Magnenat-Thalmann N. An automatic modeling of human bodies from sizing parameters. ACM SIGGRAPH 2003 Symposium on Interactive 3D Graphics. 2003. p. 19-26.

[15] Weber J. Run-time skin deformation. In: Proceedings of Game Developers Conference. 2000.

[16] Lee W.-S, Magnenat-Thalmann N. Fast head modeling for animation. Journal Image and Vision Computing 2000;18(4):355-64.

[17] Marschner SR, Guenter B, Raghupathy S. Modeling and rendering for realistic facial animation. In: Proceedings of 11th Eurographics Workshop on Rendering. 2000. p. 231-42.

[18] Feldmar J, Ayache N. Rigid and affine registration of smooth surfaces using differential properties. In Proceedings of 3rd European Conference on Computer Vision (ECCV). 1994. p. 397-406.

[19] Bookstein FL. Morphometric tools for landmark data. Cambridge University Press; 1997.

[20] Noh JY, Fidaleo D, Neumann U. Animated deformations with Radial Basis Functions. ACM Virtual Reality and Software Technology (VRST) 2000: 166-174.

[21] Smith LI. A tutorial on principal components analysis, maintained by Cornell University, U.S.A.; 2002.

[22] Arya S, Mount M, Netanyahu NS, Silverman R, Wu AY. An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM 1998;45(6):891-923.

[23] Zhu C, Byrd RH, Lu P, Nocedal J. Algorithm 778. L-BFGS-B: Fortran subroutines for large-scale bound constrained optimization. ACM Transactions on Mathematical Software 1997;23(4):550-560.

[24] Pighin F, Hecker J, Lischinski D, Szeliski R, Salesin D.H. Synthesizing realistic facial expressions from photographs. In: Computer Graphics Proceedings SIGGRAPH' 98, 1998. p.75-84.

[25] Vetter T, Blanz V. Estimating coloured 3D face models from single images: An example based approach. In: Proceedings of the 5th European Conference on Computer Vision. 1998; p.499-513.

[26] Seo H, Yeo Y.I, Wohn K. 3D body reconstruction from photos based on range scan. Edutainment 2006, LNCS 3942, 2006; p.849-860.

[27] Lee W.-S, Gu J, Magnenat-Thalmann N. Generating animatable 3D virtual humans from photographs. Compuer Graphics Forum (Proceedings of Eurographics 2000). 2000; 19(3); p.1-10.

[28] Lee W.-S. Making one structured body. VSMM 2002, organized by the international society on virtual systems and multimedia. 2002; p.558-567.

[29] Xi P, Lee W.-S, Frederico G, Joslin C, Zhou L. Comprehending and transferring facial expressions based on statistical shape and texture models. Proceedings of the 24th Computer Graphics International 2006. 2006; p.265-276.

[30] Franc V, Matas J. An extension of the component-based LDA descriptor by the generalized discriminant analysis. ISO/IEC JTC1/SC21/WG11 M8727. 2002.

[31] Heisele B, Serre T, Pontil M, Poggio T. Component-based face detection. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. 2001; p.657-662.

[32] Kim T.-K, Kim H, Hwang W, Kee S.C, Lee J.H. Component-based LDA face descriptor for image retrieval. British Machine Vision Conference (BMVC). 2002.

[33] Weyrauch B, Huang J, Heisele B, Blanz V. Component-based face recognition with 3D morphable models. IEEE Workshop on Face Processing in Video. 2004.

[34] J.L. Bentley. Multidimensional Binary Search Trees Used for Associative Searching. Communication of the ACM, 18(9), September 1975.

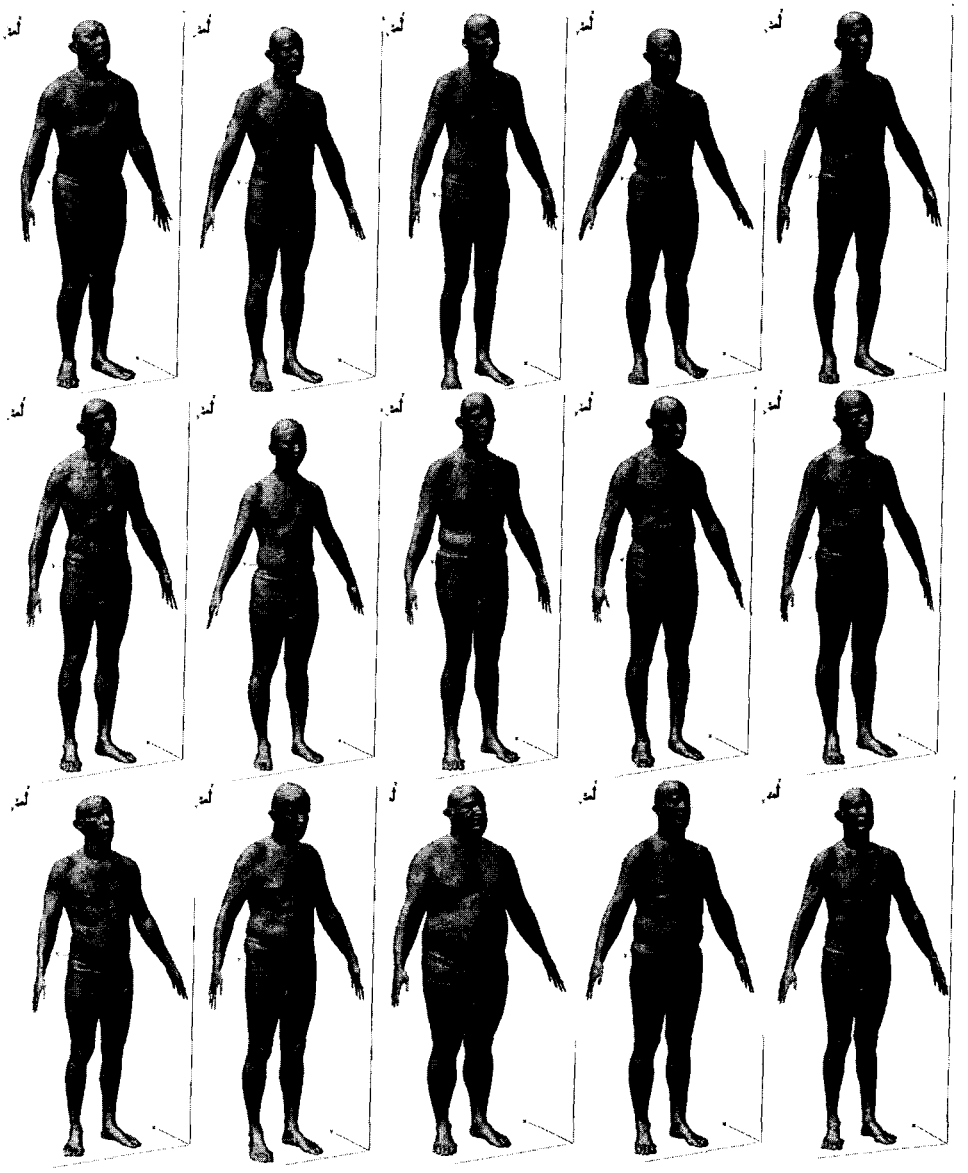
[35] Lee W.-S. Feature-based approach on animatable virtual human cloning. Ph.D. thesis, University of Geneva. 2000.

[36] Kraevoy V, Sheffer A. Template-based mesh completion. Eurographics Symposium on Geometry Processing. 2005; p.13-22.

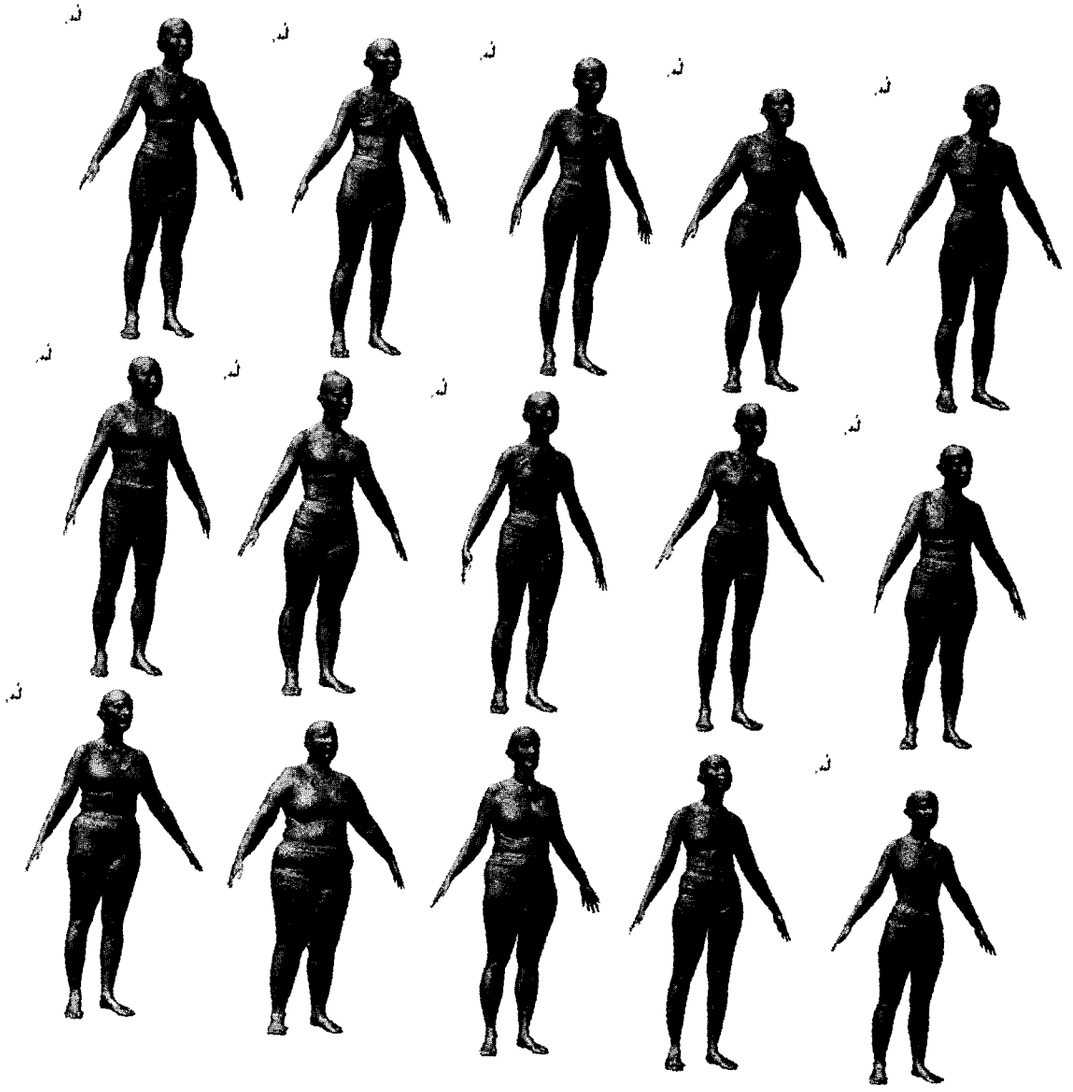
[37] Ben Azouz Z, Rioux M, Shu C. Characterizing human shape variation using 3-D anthropometric data. The Visual Computer International Journal of Computer Graphics, 22(5), 2005; p.302-314.

[38] http://en.wikipedia.org/wiki/Kd-tree

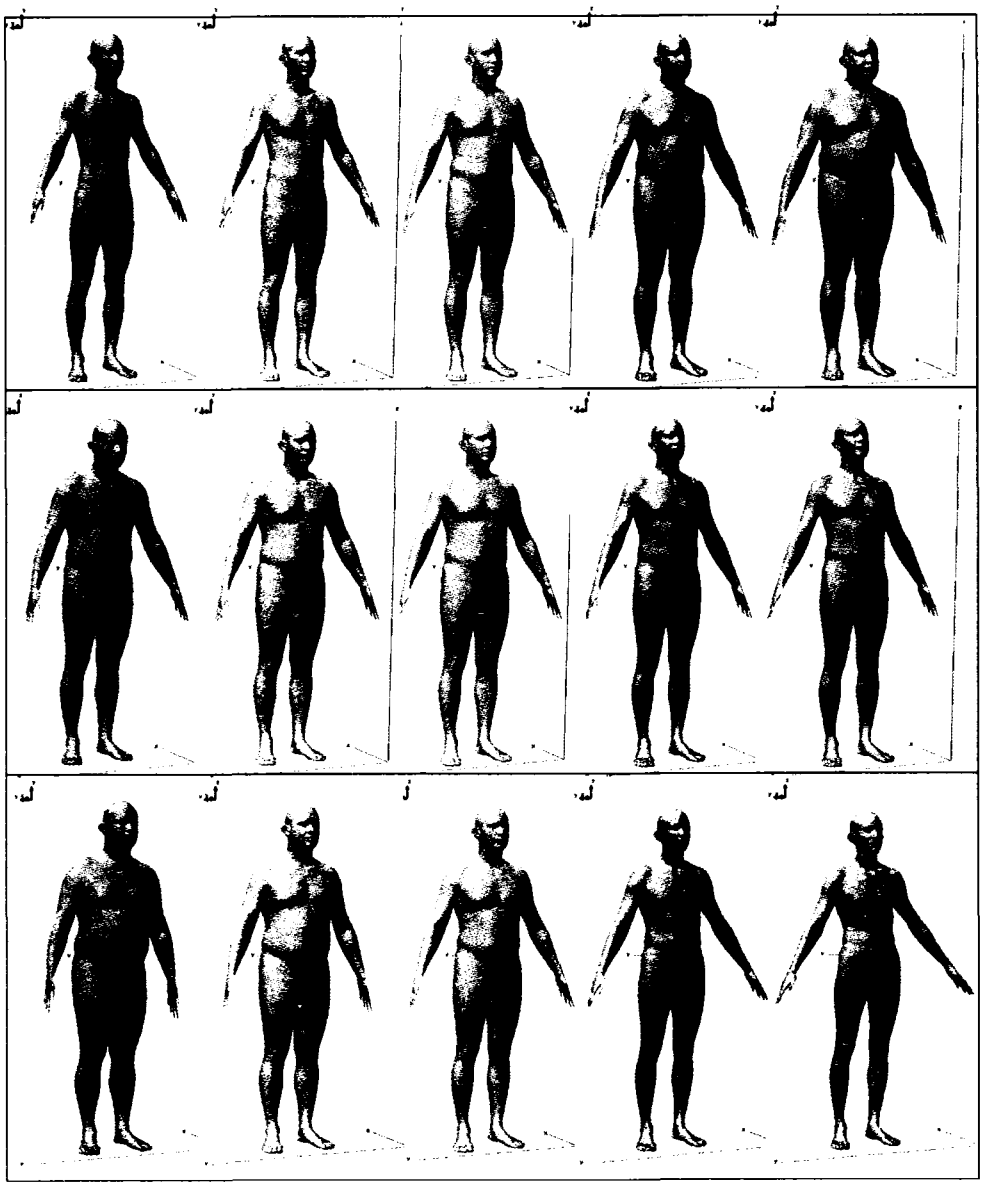
Appendix A: Samples of fine mapping results (male)



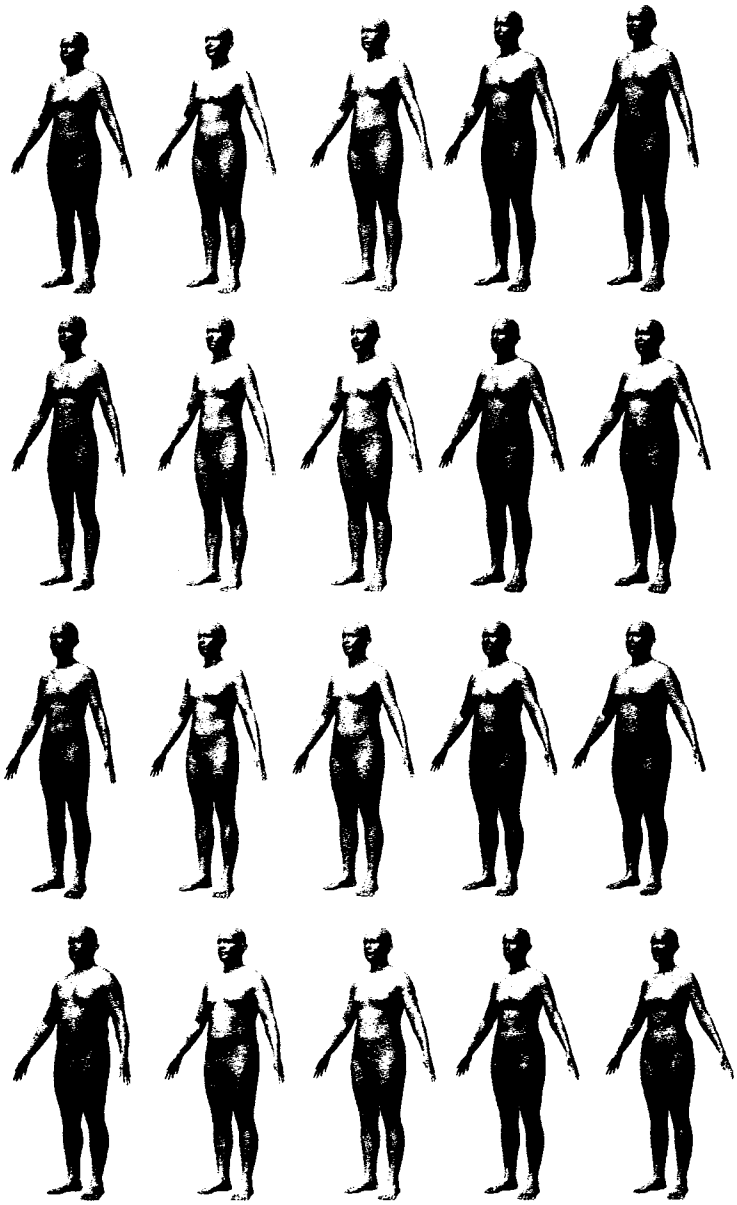
Appendix B: Samples of fine mapping results (female)



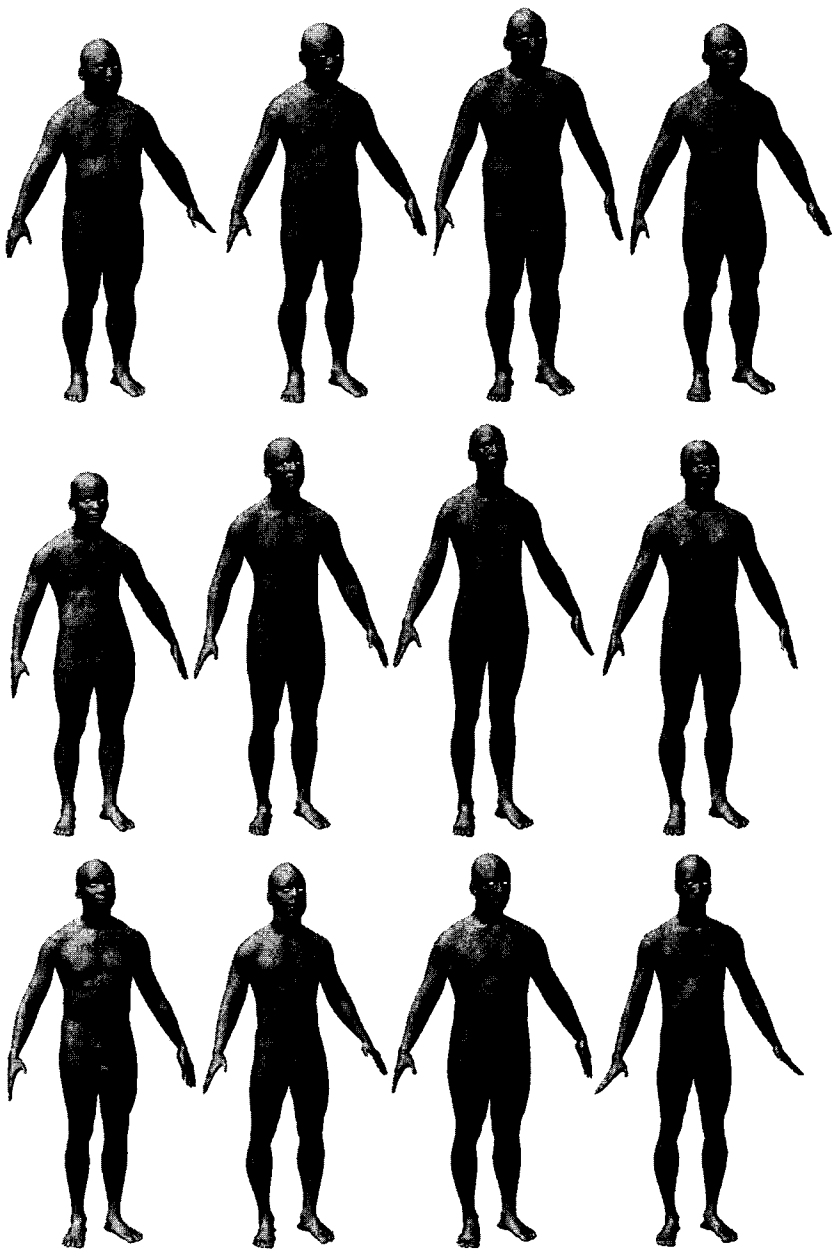
Appendix C: Pattern variations on whole body (male)



Appendix D: Pattern variations on whole body (female)



Appendix E: Results with new generic model (Male)



Appendix F: Results with new generic model (Female)