

# Generating Point Cloud from measurements and shapes based on Convolutional Neural Network – an application for building 3D human model

Nguyen Mau Tung<sup>1,2</sup>, Dang Thanh Vu<sup>3</sup>, Tran Thi Minh Kieu<sup>1</sup>, and Pham The Bao<sup>3</sup>

<sup>1</sup>University of Science and Technology, School of Textile – Leather and Fashion

<sup>2</sup>Industrial University of Hochiminh city

<sup>3</sup>Sai Gon University, Hochiminh city

## ABSTRACT

It has been widely known that 3D shape models are comprehensively parameterized using point cloud and meshes. The point cloud particularly is much simpler to handle compared with meshes, and it also contains the shape information of a 3D model. This paper, we would like to introduce our new method to generating the 3D point cloud from a set of crucial measurements and shapes of importance positions. In order to find the correspondence between shapes and measurements, we introduced a method of representing 3D data called slice structure. A Neural Networks based hierarchical learning model is presented to be compatible with the data representation. Primary slices are generated by matching the measurements set before the whole point cloud is tuned by Convolutional Neural Network. We conducted the experiment on a 3D human dataset which contains 1706 examples. Our results demonstrate the effectiveness of the proposed framework with the average error is 7.72% and fine visualization. This study indicates that paying more attention to local features is worthwhile when dealing with 3D shapes.

**INDEX TERMS** Anthropometry, 3D scanned data, Neural Network, Human body modeling, Interpolation,

## I. Introduction

A fundamental characteristic of computer-based models is capable of describing in detail the topology and geometry structure of realistic objects. 3D modeling techniques are increasingly becoming the discipline in the computer-aided design community. In addition, many applications requiring 3D models such as human animation, garment industry, and medical research have a great impact on various aspects of human life.

Although considerable research has been devoted to practicality and visualization of 3D shapes, less attention has been paired to the problem of automatically generating a 3D model. In practice, the measurement parameters like length, perimeter, and curvature have been widely used to describe the shape of realistic objects. However, reconstructing computer-based model from these measurements has still many gaps in approach. The major reason is that the set of sparse measurements fail to capture the

complex shape variations necessary for reality. On the other hand, it is impractical to resort to scanning equipment which is time-consuming and expensive.

The aim of this study is to formulate a novel representation of a 3D model based on point cloud that would make it easy to explore the relationship between the measurements and 3D shapes using Neural Networks system. Overall, our proposed framework creates the 3D point cloud when considering a set of measurements as input. Key to our approach is to separate an object into independent components and slices. This secession allows us to specifically define each architecture of Neural Network for each slice shape instead of working on a whole 3D shape. The point cloud not only has simple and unified textures compared to the diversities and complexities of mesh but also remains meaningful structure of object's boundaries and skeleton. Taking the 3D human model for an application, we here demonstrate an end-to-end procedure of synthesizing a new human model given anthropometric measurements and a set of parameters learned from training data.

## II. Related works

One of the first attempts to solve for 3D model reconstruction problem was template model based. More precisely, this method produces a new model by deforming a template model. Allen et al formulated an optimization problem to find an affine transformation at each vertex of the designed template model for fitting a 3D scanned human body. They defined three types of error and combined them to create the objective function. Their approach also dealt with incomplete surface data and filled in missing and poorly captured areas caused by the scanner [1]. Modifying the method of Allen, Hasler performed non-rigid registration with the aim of fitting pose and shape of 3D scans form a template model [2]. Seo et al deformed an existing model to obtain the new one based on two stages preprocessing: The skeleton fitting found the skeleton structure that approximates the corresponding 3D human body. The skin fitting calculated the displacement vector of each vertex between the template model after skeletal fitting and the scan mesh fitting [3].

The other approach is 2D based reconstruction. This method reduces the cost because it only requires a set of images. However, the image data often contain noises and background which are hard to remove. Blanz's approach took a human face color image as an input and generated the corresponding 3D face model. New faces and expression could be described by forming linear combinations of prototypes [4]. In their work, the weight vector was assumed to distribute as Multivariate Gaussian and could be found by maximum posterior probability. Chen attempted to automatically reconstruct more complex 3D shapes like human bodies from 2D silhouettes with the shape prior which was learned directly from existing 3D models under a framework based on GPLVM [5]. However, this approach is not realistic because of relying on the silhouettes only will cause the loss of depth information of a human body.

Most of the solutions come from the statistics based approach. Like our manner, these methods use the training set to learn the correlation between input and output, or construct an example space for extrapolation. Inspiring from DeCarlo's work [6], the statistics based model has become a powerful tool for demonstrating the feature space of the 3D model. In his studied, human face measurements were used to generate 3D face shapes by variational modeling while a prototype shape was considered as a reference. Allen reduced the dimension of 3D human meshes from

180000 elements to 40 or fewer by using principal component analysis (PCA). Then, linear regression was used as a technique to find the relationship between six different anthropometrics and 3D human model [7]. Seo defined two synthesizers which were joint synthesizer and displacement synthesizer. Joint synthesizer handles each degree of freedom of the joints, in other words, this synthesizer constructs the skeleton for the model, while another synthesizer was used to find the appropriate displacement on the template skin. These synthesizers were all learned from eight body measurements with the corresponding model by the use of Gaussian radial basis functions [8]. With the same approach to Allen's research, Chu et al attached a procedure of feasibility check to determine whether the semantic parameter values input by the user is rational. The feasibility check was based on the mathematic concept of the convex hull and if the input parameters failed the check, their system would return the most similar model in the training data [9]. Wang analyzed a human body from laser-scanned 3D unorganized points through many steps [10]. He built the feature wireframe on the cloud points by finding the key points and linking all of them with curve interpolation. After that, feature patches were generated by using Gregory patch and updated by a voxel-based algorithm. According to the introduced feature model, anthropometric measurements are easily extracted so that he used numerical optimization to generate a new 3D human body which is extracted measurements are likely to the user input sizes. Baek et al. performed PCA on both the body size and body shape vectors, then they found the weight values of the new model based on the parameter optimization problem with the constraints was the 25 user input measurements [11]. They also clustered hierarchically their shape vector space by an agglomerative cluster tree to remain small variation in each cluster. Wuhler introduced a technique that extrapolates the statistically inferred shape to fit the measurement data using non-linear optimization [12]. First, PCA is applied to produce a human shape feature space, then shape refinement is used to refine the predicted model. The objective function is formulated based on the sum of square error of three types of measurements. The author announced that the method could generate human-like 3D models with a smaller training dataset. The above methods have been suffered from a common drawback, which is the limitation of generated shapes to the space spanned by the training data. In other words, finding a large

number of variables by optimizing on the small data set would lead to the under-fitting problem.

### III. Methodology

In this section, we demonstrate our method which consists of two main steps: generating primary slices and refining 3D point cloud. 3D objects are formed by a set of planes which are perpendicular to the axial height of the object. In other words, building 3D shapes is equivalent to building all these planes. Normally, if the surfaces are smoothly divided (the distance between two adjacent planes is very small), adjacent surfaces will have nearly similar shapes. Moreover, there are no all measurements in practice thus we only considered some available ones as the measurements corresponding with primary planes. Therefore, selecting the main planes helps us to reduce the number of calculations and also necessary measurements.

Let us assume the set of all surfaces that are perpendicular to the axial height of a 3D object is  $S = \{S_i \mid i = 1 \dots m\}$ . The primary set is a subset of  $S$ ,  $PS = \{S_i \in S \mid i \in PI \subset \{1, 2, \dots, m\}\}$  such that for all  $S_i, S_j \in PS, i \neq j$ ,  $S_i$  and  $S_j$  do not have a common shape. We assessed the degree of differences of two shapes based on observing the 3D object structure. To learn the relationship between measurements and each primary surface, we construct a map from an initial set to a target set

$$f_i: C_i = \{(x, y) \in \mathbb{R}^2 \mid x^2 + y^2 \leq \left(\frac{m_i}{2\pi}\right)^2\} \rightarrow S'_i \quad (1)$$

Such that the difference of  $S_i$  and  $S'_i$  is smallest. If we consider hollow 3D objects and the surfaces turn into the slices defined in the following section,  $C$  will be a circle with its radius is computed by the perimeter of the corresponding slice.

From the principal surfaces, we can interpolate the whole 3D object since the surfaces between two principal slices whose shape gradually changing to match the shape of these two principal slices. However, the interpolated surfaces are not as practical as the actual ones. We overcame this problem by using the adjusting model that will be clarified in the next section.

#### A. Building primary slices

We restricted our study to a class of surfaces which can be written under the trigonometric formula. Representing a surface of 3D point cloud by a set of points  $S_{z_0} = \{(x, y, z) \in \mathbb{R}^3 \mid z = z_0\}$ , so that for all  $\theta \in [0, 2\pi]$  and  $r > 0$ , there is no more than one point  $(x, y, z) \in S_z$  satisfied Formula (2).

$$\begin{cases} x = x_0 + r \cos \theta \\ y = y_0 + r \sin \theta \end{cases} \quad (2)$$

Where  $(x_0, y_0)$  is former given, in this study, we called it as ‘‘anchor point’’ which is the center of a slice, Figure 1. We named the data structure defined above as ‘‘slice-structure’’.

The above surface description has an advantage that the redundancy of the third dimension is eliminated. A point  $(x, y)$  could be replaced by a pair  $(r, \theta)$ , but the  $\theta$  variables are actually in common for all slices. Thereby, a slice is written as a vector of the distances between the anchor point and points on this. Moreover, this representation is invariant under translation because of the equability of  $r$  when we translate 3D models. The rotation is also easy to handle since we merely shift the components of slice-vectors.

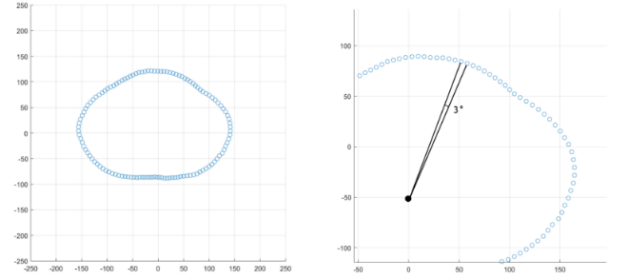


Figure 1. Example for our defined slice, each point is  $3^\circ$  apart.

Let  $PI \subset \{1, 2, \dots, m\}$  is an index set of main slices, we approximated the target slices  $S'_i, i \in PI$  by Formula (3). Let  $S_i, i = 1, \dots, m$  is the  $n$ -dimensional vector representing the  $i^{th}$  slice, the  $k^{th}$  component of it is the distance between the center and the point at  $\theta = 2\pi \frac{k-1}{n}, k = 1 \dots n$ . We defined the deformation function  $f_i: \mathbb{Z}^{+n \times 1} \rightarrow \mathbb{Z}^{+n \times 1}$  as

$$S'_i = f_i(X_i) = (W^2)^T \alpha((W^1)^T X_i) \quad (3)$$

Where  $W^1 \in \mathbb{R}^{n \times L_1}, W^2 \in \mathbb{R}^{L_1 \times n}$ ,  $\alpha$  is a non-linear function,  $X_i$  is an initial slice,  $f_i$  is also called as the Multilayer Neural Network (MNN) model.

Algorithm 1 summarizes the learning procedure of generating principal slices.

---

**Algorithm 1:** Building primary slices form measurements

---

Input: Set of measurements  $P$ , Set of 3D shapes in which  $S^{(i)} \in D$  is a sample following the slice-structure.

Output: Set of learned parameters  $W$

---

```

for  $h \in SP$  do
     $loss_h = 0$ 
    for each sample  $S^{(i)} \in D$  do
         $w = \text{length of vector } S_h^{(i)}$ 
         $r = (P_h^{(i)})/2\pi$ 
        init  $X$  as  $w$ -dimensions array with  $X_k = r$ 
         $Y = f(X)$ 
         $\max = \max Y$ 
         $\min = \min Y$ 
         $loss_h = loss_h + \frac{1}{w} \|Y - S_h^{(i)}\| +$ 
         $\left(2\pi \sqrt{\frac{(\max)^2 + (\min)^2}{2}} - P_h^{(i)}\right)$ 
     $W^{(h)} = \text{argmin}(loss_h)$ 

```

---

The key idea of the first model is to deform an initial shape into the desired shape controlled by the perimeters and the training data. Object circumferences are only useful when the object shape is revealed hence using circumferences alone to construct an object in detail is insufficient. Therefore, our approach also based on the shape of objects that can be extracted by NN model from the training set. In this work, the learning model seeks for positions on the initial slice that need to be shrunk or dilated, Figure 2.

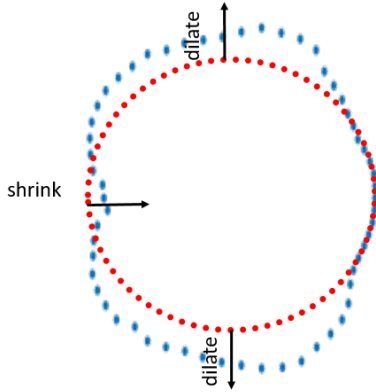


Figure 2. Deforming an initial slice (red circle) into the target slice (blue curve).

## B. Generating point cloud

Based on the results of the above step, we performed interpolation all the remaining slices. In more detail, considering  $\theta = 2\pi \frac{(k-1)}{n}$ , we calculated  $S'_{ik}, i \notin SP$  based on  $S'_{ik}, i \in SP$  and this simple task was done by linear interpolation, Figure 3. We used these interpolated slices as the input for the second model. We constructed the second synthesizer based on Convolutional Neural Network (CNN) [13] because its kernels have an ability to capture local characteristics and that is especially useful when we have to take the relationship of adjacent slices into account. This model corrected wrong interpolated points by using information on the training set via CNN architecture. The local structure of 3D shapes was retained by convolutional layers in CNN hence resulting in fine refinement. We defined our CNN model as a function  $g: \mathbb{R}^{+m \times n} \rightarrow \mathbb{R}^{+m \times n}$ , Formula (4).

$$Y = g(X) = W^L * \alpha_{L-1} \left( W^{L-1} * \alpha_{L-2} (\dots \alpha_3 (W^3 * X)) \right) \quad (4)$$

Where  $\alpha_l, l = 3, \dots, L-1$  is a non-linear activation function,  $X$  is formed by stacking both principal and interpolated slices in rows.

A rational choice of loss function for this problem is Mean Square Error (MSE). In this study, MSE calculated the difference between the generated and the actual value of each point distance on each slice. We used this metric to evaluate the error on both learning models, Algorithm 1&2. We also added the error term of perimeter into the first model's loss function.

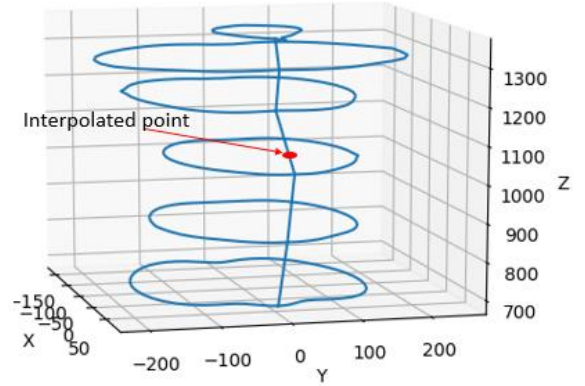


Figure 3. Interpolation of intermediate slices.

---

**Algorithm 2:** Constructing 3D point cloud

---

Input: Set of generated primary slices  $D'$  in which  $S^{(i)} \in D'$  comes from Algorithm 1, Set of 3D shapes in which  $S^{(i)} \in D$  is a sample following the slice-structure

Output: Set of learned parameters  $W$

---

$loss = 0$

for each sample  $S^{(i)} \in D$  do

$n =$  the number of rows of  $S^{(i)}$

$m =$  the number of column of  $S^{(i)}$

    init  $X$  as  $n \times m$  matrix

    for  $k$  from 1 to  $m$  do

        for  $h$  from 1 to  $n$  do

$start = \max\{x \in PI | x \leq h\}$

$end = \min\{x \in PI | x \geq h\}$

$X_{hk} = S'_{start,k} + (S'_{start,k} - S'_{end,k}) * (h - start) / (end - start)$

$Y = g(X)$

$loss = loss + \frac{1}{mn} \|Y - S^{(i)}\|$

$W = \operatorname{argmin}(loss)$

---

#### IV. An application for building 3D human model

##### A. Dataset

The datasets used in this work were independently developed by two universities in Vietnam. The following table summarizes our datasets.

Table 1. Summary of dataset

Owner	Gender /Age	The amount of avatars	The amount of damaged avatars
Industrial University of Ho Chi Minh City (IUH)	Male/ 20-60	1106	65
Hanoi University of Science and Technology (HUST)	Female/ 20-60	600	63

Each sample on both datasets was generated by the 3D scanning device and saved under '.obj' format. Each person only provided one 3D scan of the body hence the number of participants and samples is equal. Participants were suggested wearing a tight suit and

complied with the standard pose when scanning their body. We split a 3D avatar into five parts, they are torso, left leg, right leg, left arm and right arm.

In detail, these datasets were built from different devices thus they have some distinct features, Figure 4. The most noticeable thing is that the point density of the male avatars is not as dense as that of females. 3D female avatars have unified structure and each their vertex was distributed into one of five above parts. Each point on torso slices, leg slices, arm slices is 3, 5, 10 degrees respectively apart. In addition, all slices are equally spaced by the same distance in height. Meanwhile, male dataset did not meet the ideal condition like its counterpart. Not only it has no predefined boundary between two parts but also the point cloud does not follow our slice-structure. For this reason, the creator of the man dataset provided a set of landmarks for each avatar, and we used them as reference points to perform partition on the man model, Figure 5. Moreover, our slice-structure could be achieved by proper preprocessing steps.

##### B. Preprocessing

We split the whole 3D human model into five parts as the following manner, the positions mentioned below are in the landmarks set.

Torso

- Upper torso: From neck to armpit, limited by left elbow and right elbow.
- Lower torso: From armpit to hip, limited by left and right hip or limited by left and right stomach.

Arm (Left/Right)

- Upper arm: From armpit to elbow, limited by armpit and elbow.
- Lower arm: From elbow to wrist, limited by elbow and wrist.

Leg (Left/Right)

- Upper leg: From hip to knee, limited by crotch point and knee.
- Lower leg: From knee to ankle, limited by knee and ankle.

After determining all parts of a human model, we made dividing slices based on planes that perpendicular to the high axis. Let us assume that the



set of all points containing in a human part is  $S$ , we assigned

$$(x_0, y_0, z_0) := \left(x_0, y_0, \min H_z + i \frac{d_z}{m-1}\right) \quad (5)$$

If

$$\min H_z + i \frac{d_z}{m-1} \leq z_0 < \min H_z + (i+1) \frac{d_z}{m-1} \quad (6)$$

Where  $H_z = \{z \in \mathbb{R}, (x, y, z) \in S\}$ ,  $d_z = \max H_z - \min H_z$ ,  $i = 0, \dots, m-1$  and  $m$  is the number of

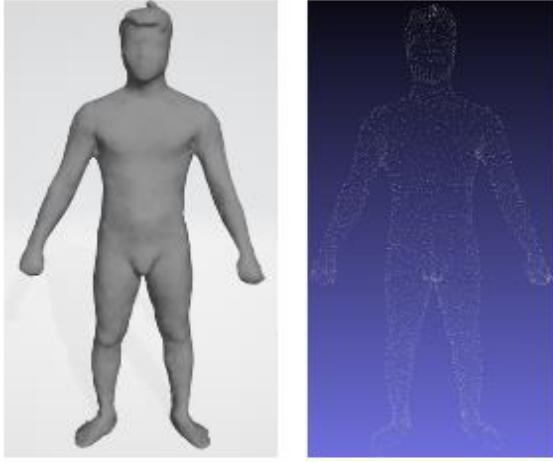


Figure 4. The avatars in male and female dataset, and its point cloud.

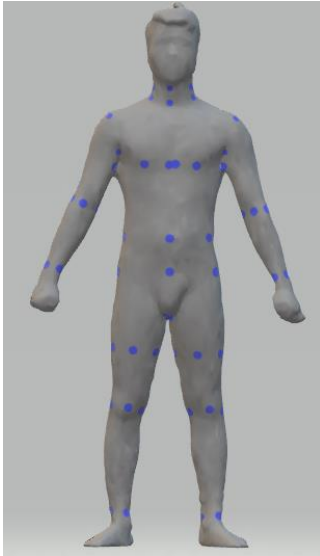


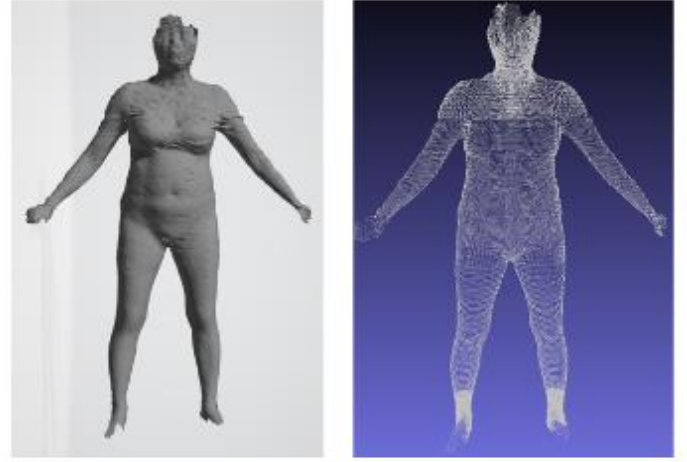
Figure 5. Some anatomical landmarks of a male model.

However, there are some downsides on the approach above. Firstly, there are some slices that the count of

slices (50 in our experiment),  $S_i = \{(x_0, y_0, z_0) | z_0 = \min H_z + i \frac{d_z}{m-1}\}$ , Figure 8.a.

The next step is to construct the slice vectors. First, we calculated the position of an anchor point on each slice. The mean formula is suitable to find these points, Formula 7.

$$(x^{(i)}, y^{(i)}, z^{(i)}) = \frac{1}{|S_i|} \sum_{(x,y,z) \in S_i} (x, y, z) \quad (7)$$



points on them is not sufficient enough to approximate the actual center point. The second thing is when constructing a new human model, we need a skeleton of it. In other words, it requires an available set of anchor points. Thanks to the landmark set, we could approximate the skeleton of male avatars. Take the torso for example, we constituted its skeleton by the line connecting the center of four neck landmarks and the crotch point, Figure 7. Once the anchor lines were found, calculating the anchor points at any height is a trivial task. The template skeleton was formed based on analyzing the position of all anchor points on the whole training dataset. In our work, we simply built the skeleton template by taking the average of the anchor points of each slice.

Given  $(x_0, y_0, z_0) \in S_i$ , the angle established by the anchor point  $(x^{(i)}, y^{(i)}, z^{(i)})$  and this point is computed by Formula (8).

$$a_0 = \arctan\left(\frac{y_0 - y^{(i)}}{x_0 - x^{(i)}}\right) \quad (8)$$

The  $j^{th}$  component of a slice vector represents the distance between the anchor point and the point at  $\theta = 2\pi \frac{j-1}{n}$ ,  $n$  is the dimension of the slice vectors. One point is distributed to the  $j^{th}$  position if satisfying the following condition.

$$2\pi \frac{j-1}{n} \leq a_0 < 2\pi \frac{j}{n} \quad (9)$$

Where  $j = 1, \dots, n$ . The distance is directly calculated by Euclid metric, Formula (10).

$$d_j^{(i)} = \sqrt{(x_0 - x^{(i)})^2 + (y_0 - y^{(i)})^2} \quad (10)$$

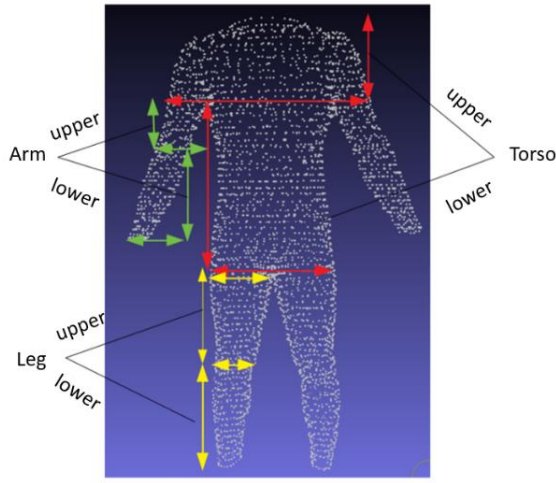


Figure 6. 3D human model partition.

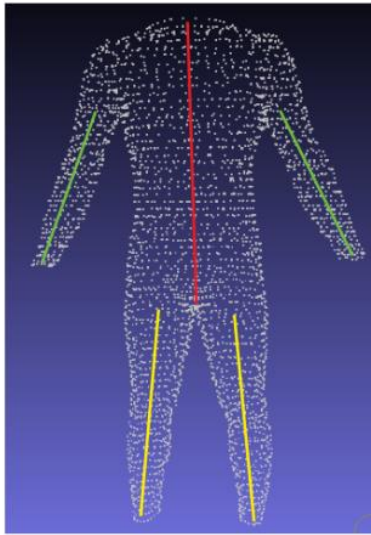


Figure 7. the anchor lines of a human model.

Both man and female avatars suffer from missing data problem. In the female dataset, the reasons are the carelessness during the scanning process and the outdated equipment. On the other hand, the missing value issue on preprocessed male models is inevitable because their original point cloud is not ideal. Furthermore, the point density is not sufficiently dense to divide the male body into many slices. We tackled this problem by performing linear interpolation on grid data of slice vectors, Figure 8.b.

### C. Measurements

The male dataset supplied us a set of anthropometric measurements with 178 categories comprising slice perimeters, width, and height of body parts. Nevertheless, the measurements are not in the same unit with distances computed on the point cloud. Meanwhile, the female dataset provided no measurements. Due to these reasons, we decided to recalculate the measurements to be consistent in both datasets. The simple way to compute a slice circumference is summing all distances of two adjacent points, but it does not seem realistic when measuring non-convex shapes. We proposed using the circumference of the convex hull of a slice as its measurements. These sizes were calculated on the primary slices, Figure 9.

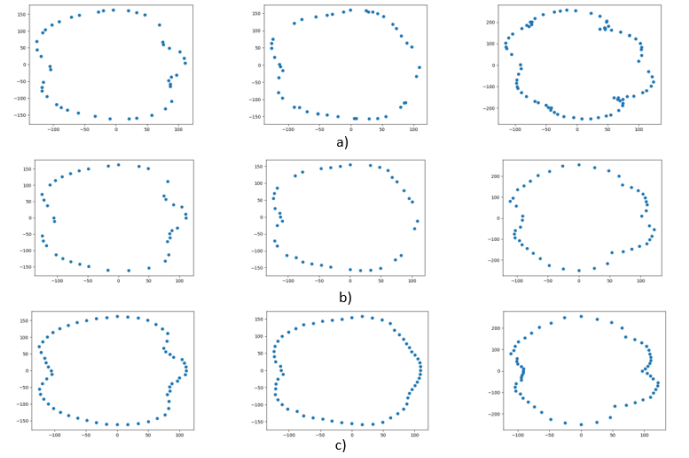


Figure 8. Preprocessing step

a) original slices, b) after selecting slices components, c) after filling missing data.

Summary, there are 28 slice measurements, but we can reduce the number of measures to 17 because of the similarity of the right and left side. In addition, it is necessary to record the height (length) of each body to

entirely build up the 3D human model. This would lead to 20 measurements in total. The primary positions were chosen based on the statistics on the dataset and the standard ratio of the human body [14]

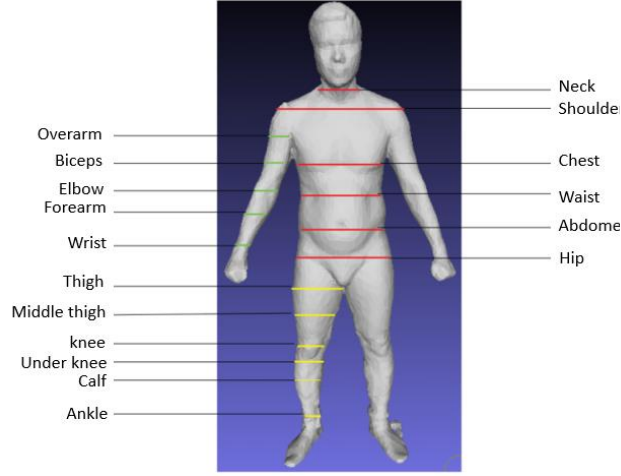


Figure 9. The position of primary slices.

#### D. Learning model

To construct primary slides, we built Neural Network (NN) models with one hidden layer as described in section III.A. These model deformed input slices into target slices, Figure 10.

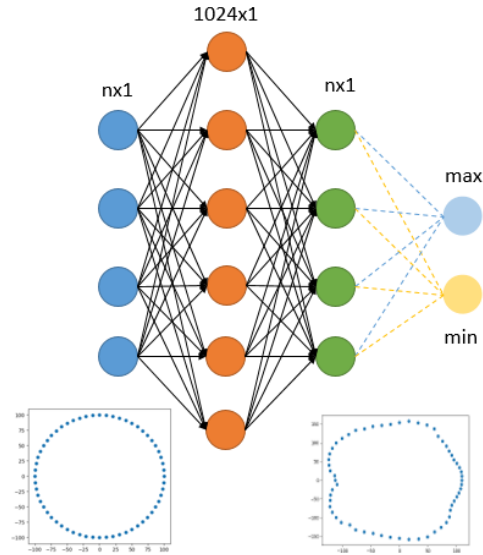


Figure 10. The Neural Network model for generating the primary slices.

These models take an initial circle as input and learns the deformation from the input shape into the target shape. The radius of the initial circle is  $r = \frac{p}{2\pi}$ , where  $p$  is the perimeter of the slice being considered. The

input and output size depends on the body part, in our experiment,  $n$  equal 20, 30, 60 for the arm, leg, torso, respectively. The error between a predicted and the actual slice is calculated by Formula (11).

$$L(y, y') = \frac{1}{n} \sum_{i=1}^n (y_i - y'_i)^2 + \gamma \left( 2\pi \sqrt{\frac{(\max y')^2 + (\min y')^2}{2}} - p \right) \quad (11)$$

Where the second error term comes from the difference between the approximation of the circumference of predicted slice and the actual one. The objective function reflects the error not only at each component (local information) but also the perimeter of the slice (global information).

Once the entire main slices had been found, linear interpolation was used to infer all remaining slices. These interpolated slices are the input to the second NN model as described in section III.B, Figure 11. We used ReLU [15] as the activation function in both architectures.

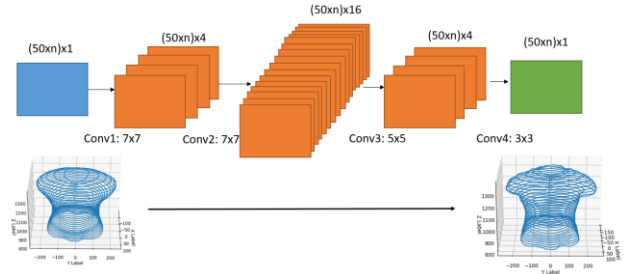


Figure 11. The Neural Network model for adjusting entire slices.

The convolutional layers help the model to learn the local correlation of adjacent slices. As a result, the remaining slices would be corrected based on the primary slices. The most cautious thing when building CNN for this problem is padding. To remain the size of data when transferring through multiple layers, we performed reflex padding in vertical and symmetric padding in horizontal, Figure 12. Padding this way retains circularly linked characteristic of slices.

		symmetric					
reflex	3	2	1	2	3	1	2
	3	2	1	2	3	1	2
	6	5	4	5	6	4	5
	9	8	7	8	9	7	8
	9	8	7	8	9	7	8

Figure 12. Padding strategy



We defined the loss function on this second model by using MSE, Formula 12.

$$L(y', y) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (y_{ij} - y'_{ij})^2 \quad (12)$$

Where  $y, y'$  are respectively a matrix of the actual and predicted distances to the anchor point of a body part.

## V. Experiments and results

We trained our NN models on the Linux server with 24 GB RAM, GPU with 12 GB RAM, and Xeon CPU with 2.2Ghz. We used Python as the implementing language and the main libraries using in our experiment are pytorch and numpy. We used Adam algorithm [16] to minimize the objective function, the meta parameters were set according to recommendation of the authors (learning rate  $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$ ). We evaluated the error by the average relative error, Formula 13.

$$L(y, y') = \frac{1}{mn} \sum_{i=1}^n \sum_{j=1}^m \frac{|y_{ij} - y'_{ij}|}{y_{ij}} \times 100\% \quad (13)$$

Where  $y, y'$  are respectively a matrix of the actual and predicted distances to the anchor point of a body part. The above error formula is not affected by the heterogeneity in size on different body parts and also on different datasets. In the male dataset, we used 1066 samples as training data and 100 samples as testing data, while 500 and 100 as training and testing data in the female dataset, the samples were selected randomly. Table 2 shows the average error on each primary slice after training 1000 epochs on the male and female dataset.

Learning the relationship between the size and corresponding slice shape is a hard problem because of the curse of dimension. Despite the fact that the input is just a scalar, we have to predict the slice vector with at least 20 components. To solve this problem, we used initial shapes. The initial shape not only is a rough approximation for the target slice but also helps the NN model increase the number of parameters and avoid under-fitting. In our work, we limited the class of initial shapes to circles that their radius is calculated by the slice perimeters. Geometrically, the first NN models act as a figure deformation controlled by the slice sizes. The NN models are the non-linear transformations from straight lines to the particular “slice vector curves” which are the slice shapes after

converting into the slice vector representation. These curves have analogous shapes if they are placed at the same position, Figure 13.

Table 2. Average error on each principal slice on the training data of male and female dataset (full dataset)

Slice	Train/Male (%)	Train/Female (%)
Hip	8.39	4.68
Abdomen	4.41	3.46
Waist	5.34	4.90
Chest	5.80	11.25
Shoulder	6.18	9.37
Neck	13.55	21.61
Left Wrist	7.77	10.99
Left Forearm	5.53	7.48
Left elbow	4.21	7.72
Left Biceps	6.22	8.80
Left Overarm	10.54	15.66
Right Wrist	13.43	8.12
Right Forearm	12.69	7.63
Right Elbow	8.36	7.01
Right Biceps	8.84	6.69
Right Overarm	12.20	12.64
Left Ankle	7.93	7.38
Left Calf	7.73	4.62
Left Under knee	11.80	3.30
Left Knee	6.31	4.40
Left Middle Thigh	3.63	4.72
Left Thigh	6.46	9.67
Right Ankle	9.54	6.64
Right Calf	7.52	4.66
Right Under Knee	10.67	3.09
Right Knee	5.80	4.45
Right Middle Thigh	3.41	5.05
Right Thigh	6.03	10.21

In the Torso part, the neck slices have the highest average error due to these slices are not clearly separated from the head and the anatomical landmarks at neck position are placed at the wrong locations like collar or chin. This reason leads to that the shape of the neck slices varies considerably. The same thing happens with overarm slices. The boundaries between arms and shoulder are not accurately determined based on the landmarks. Another problem is the lack of a large number of components on overarm slice vectors because of the obstructed locations such as armpits which are ignored by the 3D scanner, Figure 14.

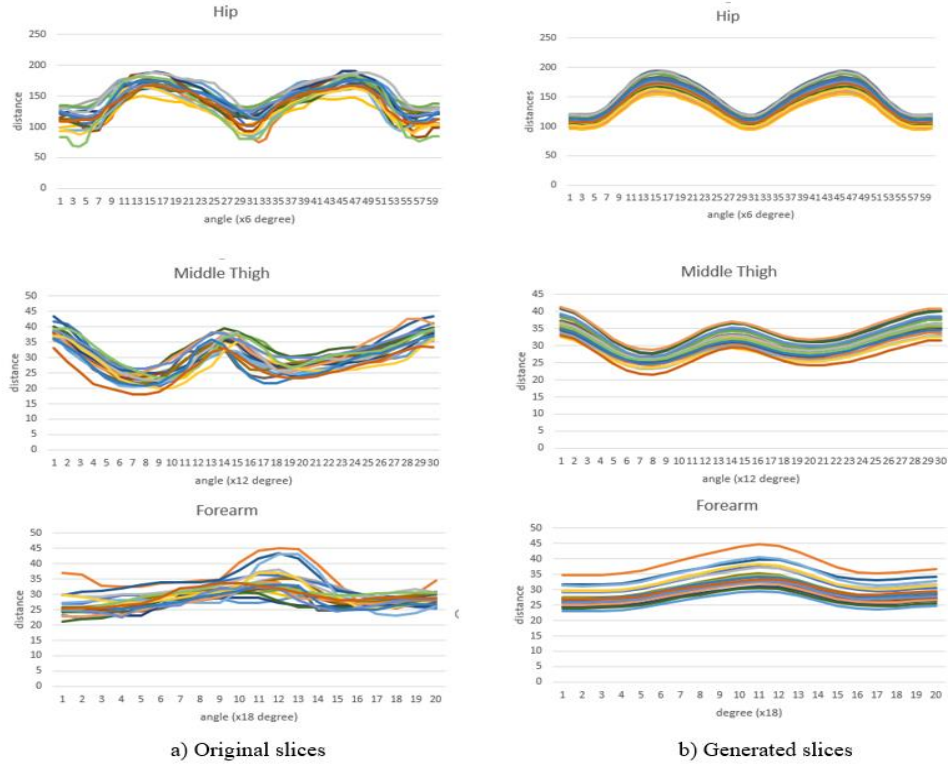


Figure 13. The slice vector “curves” of wrist, hip and thigh of 20 examples in male dataset.

Table 3. Average error on each part of a 3D male and female model after activating the CNN model (the test set is comprised of damaged and undamaged samples).

Part	Train/ Male (%)	Train/ Female (%)	Test/Male (undamaged) (%)	Test/ Female (undamaged) (%)	Test/Male (damaged) (%)	Test/Femal e(damaged) (%)	Test/Male (average) (%)	Test/ Female (average) (%)
Torso	7.42	11.26	6.46	9.74	10.73	15.79	8.59	12.765
Left arm	8.24	15.20	6.68	15.83	10.87	26.11	8.77	20.97
Right arm	12.99	12.23	10.83	12.33	14.72	13.82	12.77	13.075
Left leg	8.78	7.26	7.59	7.40	11.22	9.92	9.40	8.66
Right leg	8.39	8.01	7.81	7.45	11.81	13.26	9.81	10.35
<b>Average</b>	<b>9.16</b>	<b>10.59</b>	<b>7.87</b>	<b>10.55</b>	<b>11.87</b>	<b>15.78</b>	<b>9.86</b>	<b>13.16</b>

Table 3 shows the result after training the CNN models to entirely construct a full human body. To conduct this section, we also used Adam algorithm with 1000 epochs. We chose 50 good samples and 50 damaged samples to form the test set. Thus we could evaluate the influence of bad patters on the overall test accuracy. The results show that the errors in the undamaged test sets are approximate to the training errors. On the other hand, the errors in the damaged test sets are not as low as the good ones. Based on the

results, we can conclude that our framework is non-sensitive to the small amount of damaged samples. Moreover, the amount of sample in the training set is sufficient to do inference on the shape of testing samples.

While analyzing the database, we realized that there are many damaged samples in both datasets. The issues in the female dataset almost come from the scanning device. While the problems in the male dataset are due to the non-cooperation of participants,

Figure 15. We eliminated all unqualified samples from both datasets. Overall, there are 65 samples in the male dataset and 63 samples in the female dataset. After removing these patterns, we conducted a new training procedure on the new training and testing sets, the results are shown in Table 4. In the male dataset, there are 1001 training samples and 100 testing samples, while there are 437 and 100 samples as training and testing data in the female dataset. The average errors after feeding the interpolated primary slices into the CNN models are lower than the errors of themselves when compared to the ground truth.

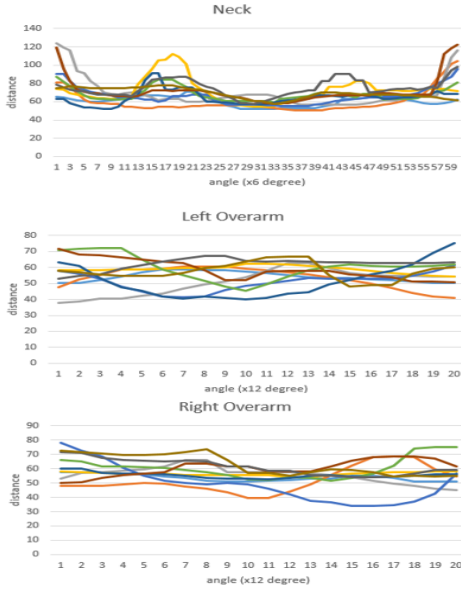


Figure 14. The slice vector “curves” of neck, left and right overarm of 10 examples in the male dataset

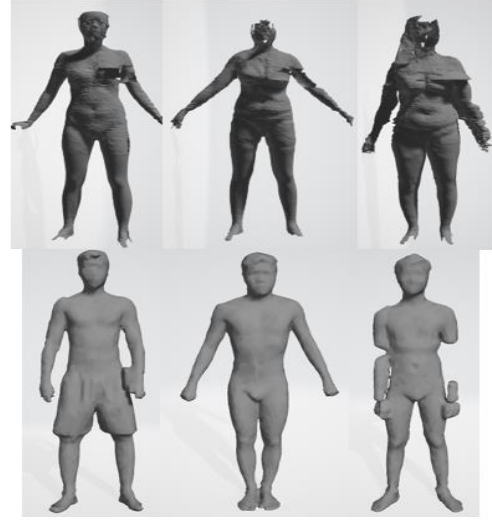


Figure 15. Examples for the damaged 3D human model.

The average training and testing time per body part are shown in Table 5.

Table 5. Average training and testing time on both datasets.

Part	Train (s)	Test a sample (s)
Torso (primary x6)	474	0.013
Torso (all)	1484	0.121
Arm (primary x 10)	540	0.032
Arm (all)	133	0.098
Leg (primary x12)	780	0.035
Leg (all)	764	0.108
<b>Total</b>	<b>4175</b>	<b>0.407</b>

Table 4. Average error on each part of 3D a male and female model on the undamaged datasets before and after activating the CNN model.

	Train/ Male (Before)	Train/ Female (Before)	Test/ Male (Before)	Test/ Female (Before)	Train/ Male (After)	Train/ Female (After)	Test/ Male (After)	Test/ Female (After)
Torso	6.90	8.04	7.21	8.27	6.68	7.13	7.94	7.94
Left arm	7.62	12.77	8.31	13.81	7.15	12.60	7.67	13.83
Right arm	8.14	10.81	8.27	10.94	7.15	10.70	7.35	10.72
Left leg	7.02	6.32	7.74	6.51	7.31	5.77	7.90	5.96
Right leg	7.53	6.63	8.12	7.08	6.90	5.89	7.39	6.13
<b>Average error</b>	<b>7.42</b>	<b>8.91</b>	<b>7.93</b>	<b>9.32</b>	<b>7.03</b>	<b>8.41</b>	<b>7.65</b>	<b>8.91</b>

Once all necessary slices are ready to build a 3D model, we perform re-meshing counted on the triangular mesh method. This simple rule constitutes a mesh by using three points. The points at  $(i, j)$ ,  $(i, j + 1)$ ,  $(i + 1, j)$  on two adjacent slices would form a mesh. Likewise, the points at  $(i + 1, j)$ ,  $(i, j + 1)$ ,  $(i + 1, j + 1)$  would also produces a mesh, Figure 16.

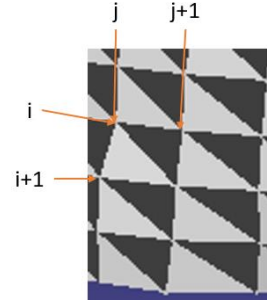


Figure 16. Re-meshing based on triangular mesh.

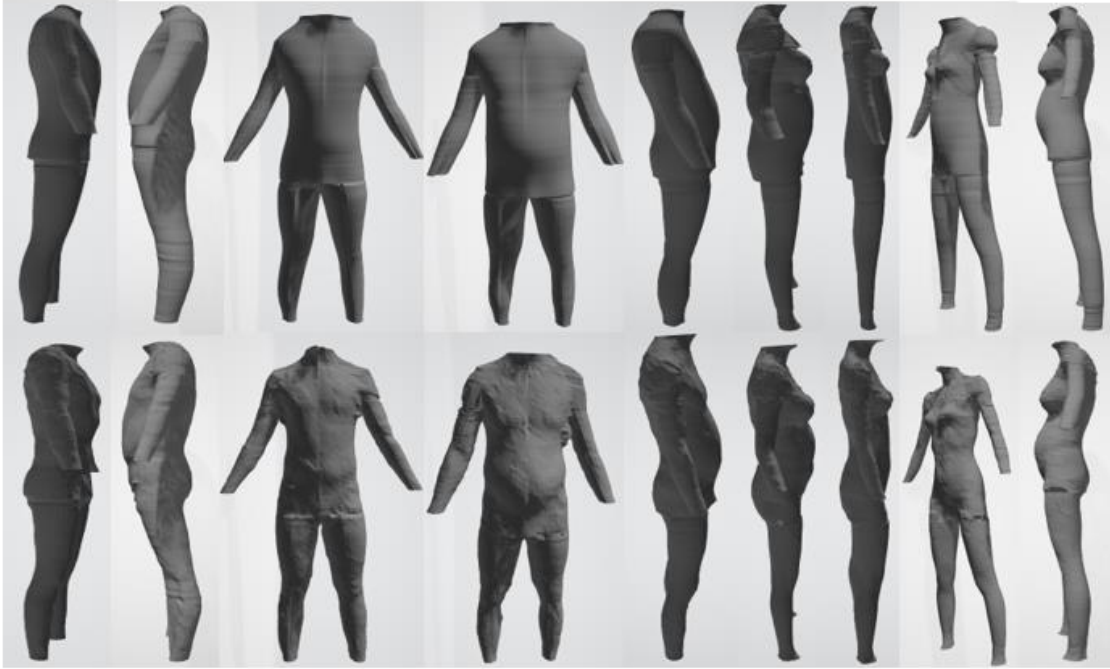


Figure 17. The 3D avatars of male and female: First row: generated shapes, Second row: original shapes.

## VI. Discussion and conclusion

Generating 3D models has been becoming an attractive field in recent years. There is no doubt about the versatility of the 3D model in computer graphics applications such as gaming, films, and garments. However, constructing a 3D shape is not a trivial task since the complexity of the model usually demands careful design, the power of computer hardware and modern scanning devices. To tackle this problem, we introduced a novel method to create a new 3D model simply by taking the measurements as input. Our main contributions include: (1) describing a formula to represent 3D data under slices of point cloud, (2) introducing two-step framework based on Neural

Networks for generating the primary slices and impaling entire slices, (3) conducting the experiment and unveiling a benchmark on the IUH and HUST 3D human dataset.

It is difficult to compare the present study's finding with other previous studies because of the different dataset and evaluating metrics. However, the results confirm the effectiveness of our approach due to the generated 3D point cloud models are fine enough for visualization with the small error during the rational running time, Figure 17. Our proposed framework not only explores the correlation between the shape and the size of a human body but also captures the local information among adjacent slices. Instead of directly inferring a whole 3D model, we divided the objective

model into specific parts and defined suitable NN architecture for each part. In the spirit of learning detail slice shapes rather than learning overall structure, the hierarchical learning strategy was introduced in which the shape of slices corresponding to user-defined measurements are the foundation of all other slices shape.

The slice structure that we used in this study is not restricted in the static case. It is also effective when applying to 3D dynamic models via a morphable skeleton. The key idea to generate a new slice shape is to deform an initial shape depending on the training dataset. Because every single step of our method is no need to change the coordinate or reduce the dimension, we ensure that a generated point cloud still looks like the samples in the training data. The main drawback of our approach is data deficiency. We suffer from the

under-fitting problem hence the NN systems cannot achieve the ideal generalization. Another problem is that we concentrate on construct a point cloud but mesh. Therefore, any application requiring 3D models with full mesh reconstruction might need more processing steps. Although slice structure is very simple, it is challenging to achieve its status, especially when disjointing 3D shapes with complex designs.

In conclusion, this study suggested that a 3D point cloud could be constructed completely when giving a set of essential measurements. On the other hand, it is necessary to consider the shape in more detail when dealing with complicated 3D structures such as human bodies. Our proposed framework shed the light on this concern since it has the ability to analyze local shape features.

### Acknowledgement

The 3D human datasets are sponsored by Industrial University of Ho Chi Minh City (IUH) and Hanoi University of Science and Technology (HUST), Vietnam. We are greatly appreciated these agents who provided us the information of these datasets. Especially, we would like to say thanks to Faculty of Garment Technology and Fashion Design, Industrial University of Ho Chi Minh City for the powerful equipment which helps us to complete this study.

### References

- [1] B. Allen, B. Curless and Z. Popovic, "The space of human body shapes: reconstruction and parameterization from range scans," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 587-594, 2003.
- [2] N. Hasler, C. Stoll, M. Sunkel, B. Rosenhahn and H.-P. Seidel, "A Statistical Model of Human Pose and Body Shape," *Computer Graphics Forum*, pp. 337-346, 2009.
- [3] H. Seo and N. T. Magnenat, "An automatic modeling of human bodies from sizing parameters," *Proceedings of the 2003 symposium on Interactive 3D graphics*, pp. 19-26, 2003.
- [4] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp. 187-194, 1999.
- [5] Y. Chen and R. Cipolla, "Learning shape priors for single view reconstruction," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, Kyoto, Japan, 2009.
- [6] D. DeCarlo, D. Metaxas and M. Stone, "An anthropometric face model using variational techniques," *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 67-74, 1998.
- [7] B. Allen and B. Curless, "Exploring the Space of Human Body Shapes: Data-driven Synthesis under Anthropometric Control," *Digital Human Modeling for Design and Engineering Symposium*, pp. 1-4, 2004.
- [8] H. Seo and N. T. Magnenat, "An example-based approach to human body manipulation," *Graphical Models*, vol. 66, no. 1, pp. 1-23, 2004.



- [9] C. H. Chu, Y. T. Tsai, C. C. L. Wang and T. H. Kwok, "Exemplar-based statistical model for semantic parametric design of human body," *Computers in Industry*, vol. 61, no. 6, pp. 541-549, 2010.
- [10] C. Wang, "Parameterization and parametric design of mannequins," *Computer Aided Design*, vol. 37, no. 1, pp. 83-98, 2005.
- [11] S. Y. Baek and K. Lee, "Parametric human body shape modeling framework for human-centered product design," *Computer-Aided Design*, vol. 44, no. 1, pp. 56-67, 2012.
- [12] S. Wuhner and C. Shu, "Estimating 3D human shapes from measurements," *Machine Vision and Applications*, vol. 24, no. 6, pp. 1133-1147, 2013.
- [13] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278-2324, 1998.
- [14] R. Davis and Altervagt, "Golden mean of the human body," *Fibonacci Quarterly*, vol. 17, pp. 340-344, 1979.
- [15] V. Nair and G. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," 2010.
- [16] D. Kingma and B. Jimmy, "Adam: A Method for Stochastic Optimization," in *3rd International Conference for Learning Representations*, San Diego, 2015.