

## Building 3-D Human Data Based on Handed Measurement and CNN

Bich Nguyen<sup>1</sup>, Binh Nguyen<sup>1</sup>, Hai Tran<sup>2</sup>, Vuong Pham<sup>1</sup>, Le Nhi Lam Thuy<sup>1</sup> and Pham The Bao<sup>1,\*</sup>

<sup>1</sup>Faculty of Information Science, Sai Gon University, Ho Chi Minh, 70000, Vietnam

<sup>2</sup>Faculty of Information Technology, University of Education, Ho Chi Minh, 70000, Vietnam

\*Corresponding Author: Pham The Bao. Email: ptbao@sgu.edu.vn

Received: 08 March 2022; Accepted: 01 June 2022

**Abstract:** 3-dimension (3-D) printing technology is growing strongly with many applications, one of which is the garment industry. The application of human body models to the garment industry is necessary to respond to the increasing personalization demand and still guarantee aesthetics. This paper proposes a method to construct 3-D human models by applying deep learning. We calculate the location of the main slices of the human body, including the neck, chest, belly, buttocks, and the rings of the extremities, using pre-existing information. Then, on the positioning frame, we find the key points (fixed and unaltered) of these key slices and update these points to match the current parameters. To add points to a star slice, we use a deep learning model to mimic the form of the human body at that slice position. We use interpolation to produce sub-slices of different body sections based on the main slices to create complete body parts morphologically. We combine all slices to construct a full 3-D representation of the human body.

**Keywords:** 3-D human model; deep learning; interpolation

### 1 Introduction

Recently, applying 3-D printing [1,2] in fashion has attracted attention but is limited to printing parts of body. This limitation has resulted in inadequate product size, and the printing price is higher than the clothes, which are being made traditionally. Particularly, in the sewing factory, they only can sew to international standard measurements. However, clothes are out of shape if the human shape is different from the standard measures.

Anthropometric surveys toward applications in the garment industry were conducted early by nations worldwide but were primarily performed by traditional hand measurement. Japan was the first country to undertake a large-scale countrywide survey using 3-D body scanners from 1992 to 1994. Other nations, in turn, performed anthropometric surveys and built their own body measuring systems utilizing 3-D body scanning technologies.

Many research groups have conducted building 3-D human models, from building a part of the human body (ears, human organs) to constructing the entire body to serve different needs such as health care, fashion, etc. Thanks to the quick growth of technology and skill, many methods of building



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

3-D human models have been improved. They are being applied in modern printing and scanning 3-D equipment to respond demands of building complete human models. Kieu et al. [3] applied regression and interpolation to build 3-D human model. They divided the model into parts, then used Hermite interpolation to interpolate each part of the model. Nguyen et al. [4] used Convolutional Neural Network to create a set of points of 3-D human model. This method used the diameter of a circle to create key slices of the model and used interpolation to create remaining slices. Zeng et al. [5] followed the anthropometric modeling approach. They proposed a local mapping technique based on new feature selection, allowing automatic anthropometric parameter modeling for each body aspect.

Point cloud learning has been gaining more attention recently due to its effectiveness in many applications such as computer vision, autonomous vehicles, and robotics. Deep learning has dramatically solved visual problems [6]. 3-D data can often be represented in various formats, including depth images, point clouds, meshes, and volumetric. As a commonly used format, the point cloud representation preserves the original geometry information in 3-D space without any changes.

In this study, we propose a new method based on deep learning and point cloud learning to conduct training of 3-D models. First, we convert the 3-D model into point clouds: Get the uniform point per slice with a probability proportional to the surface area, using the built-in function in the `pytorch3D` library. This step converts the input models into a set of model point clouds to form a complete mesh. Then, we move the input cloud points such that all of these points come close to the hand measurement data. Finally, we construct a 3-D model of the person corresponding to the data. We repeat those steps until all the cloud points have the lowest error compared to the hand-measured data.

We train our method on the 3-D model set and the model surface. Contrary to previous approaches, which need a lot of data and need to calculate the interpolation in detail, our point cloud learning method has a lower error and better control over input data and data when training the model. Also, our model has improved how to train the model on the whole 3-D model.

## 2 Relate Work

### 2.1 3-D Models

3-D models are being applied in many types of industries. The health industry uses sophisticated agency models made of many slices of two-dimensional pictures from a Magnetic Resonance Imaging or Computed tomography scan to create vivid and realistic scenes. These models are used in video game business as assets for computers and video games. In recent decades, the Earth Science community has begun to use 3-D geological models as a common practice. Physical devices such as 3-D printers or Computer Numerical Control machines can also be formulated based on 3-D models [7].

Files with the extension `.obj` are commonly used to save 3-D models. These files provide 3-D coordinates, texture maps, polygon faces, and other object information. The first character of each line in the `.obj` file is always an attribute type, followed by arguments. The common attributes in 3-D models are `v`, `vn`, `f`, `vp`, and `vt`. In this study, we use three attributes including `v`, `f`, and `vn`.

### 2.2 Deep Learning

Deep learning is a form of machine learning that uses a collection of algorithms to decode high-level model data utilizing many processing layers with structural combinations or by integrating many additional nonlinear factors. Deep learning comprises several methods, each of which has its own set of applications to the problem [8,9]:

- Deep Boltzmann Machines (DBM)
- Deep Belief Networks (DBN)
- Convolutional Neural Networks (CNN)
- Stacked Auto-Encoders

Although they may also incorporate propositional formulae or latent variables arranged by layers, most current deep learning models are based on artificial neural networks, especially CNN. Nodes in DBN and DBM are examples of deep growth models [10].

In deep learning, each level learns to transform the input data into a slightly more abstract and synthetic representation. In the application of image recognition, input is a matrix of pixels. The first layer of a CNN model encodes edges. The second layer can compose and encode the alignment of the edges. The third layer can encode the nose and eyes. The fourth layer can recognize that the image has a face. Deep learning can learn critical features to place at what level optimally [6,11–13].

An artificial neural network consists of a group of artificial neurons (nodes) connecting and processing information by passing connections and computing new values at the nodes (connectionism approach to computation). In many cases, an artificial neural network is an adaptive system that changes its structure based on external or internal information flowing through the network during the learning process.

### 2.3 Error

To calculate errors of 3-D model, we use the method in [3] to calculate perimeter base on  $i^{th}$  slice from model, as shows in Eq. (1).

$$P_i = \sum_{j=1}^{n-1} \sqrt{(x_{i,j} - x_{i,j+1})^2 + (y_{i,j} - y_{i,j+1})^2 + (z_{i,j} - z_{i,j+1})^2} + \sqrt{(x_{i,n} - x_{i,1})^2 + (y_{i,n} - y_{i,1})^2 + (z_{i,n} - z_{i,1})^2} \quad (1)$$

where

$n$  is the number of points in slice  $i^{th}$ .

$(x_{i,j}, y_{i,j}, z_{i,j})$  are the coordinates of the  $j^{th}$  point at slice  $i^{th}$ .

Error of slice is calculated using Eq. (2).

$$es = \frac{\sum_{i=1}^n (p_{gt} - p_p)^2}{n} \quad (2)$$

where

$p_{gt}$  is the perimeter of the real data scanned by the 3-D scanner.

$p_p$  is the perimeter of the 3-D model slice that we collect from the hand measurements.

$$loss = loss_{chamfer} \times w_{chamfer} + loss_{edge} \times w_{edge} + loss_{normal} \times w_{normal} + loss_{laplacian} \times w_{laplacian} \quad (3)$$

The error is calculated based on 3-D model after training and 3-D output model, using Eq. (3). where

$loss_{chamfer}$ ,  $loss_{edge}$ ,  $loss_{normal}$ ,  $loss_{laplacian}$  is Chamfer, edge, normal, and Laplacian errors, respectively.

$w_{chamfer}$ ,  $w_{edge}$ ,  $w_{normal}$ ,  $w_{laplacian}$  is Chamfer, edge, normal, Laplacian weights, respectively.

### 3 The Proposed Model for Creating a 3-D Representation of the Human Body

To build a 3-D human model, we deform a 3-D input model to create a target model according to hand measurements. To turn the input model into a target (according to the problem requirements) in each iteration, we move point in the set of points of the model from the original input model in order to build the output model so that the points of the input model get closer to the corresponding point of the output model . After finishing iterations, the distance between the points on the input model and the corresponding points on the output model will be reduced.

We set up an Algorithm 1 to make the meshes structure and algorithm 2 to build the train model. Based on Algorithm 1 and Algorithm 2, we create a 3-D model of the human body by Algorithm 3. In this article, we use deep learning to transform the original model. We choose two input 3-D models to apply the proposed method, including the 3-D sphere model (3.1) and 3-D standard human model (3.2).

---

#### Algorithm 1: Describe Input and Output Mesh

---

**Input:** 3-D model

**Output:** a meshes structure

**verts, faces, aux** = the corresponding properties of 3-D model

**faces\_idx** = faces.verts\_idx.to(device)

**verts** = verts.to(device)

**center** = (0, 0) verts = verts-center

**scale** = max(vert.abs().max(0)[0])

**verts** = verts/scale

**trg\_mesh** = point clouds are created by verts and faces\_idx

**src\_mesh** = point clouds are created by input 3-D model

**End**

---



---

#### Algorithm 2: Build the Train Model

---

**Input:** point clouds-src, trg; training times-n.

**Output:** the set of point clouds after training

The number of vertice of src.

Optimized by Stochastic Gradient Descent.

**For** i=0 to n

    Initialize optimizer

**new\_src\_mesh** = deform the mesh of deform\_verts

**sample\_trg, sample\_src** = sample points from the surface of each mesh of trg, src

    optimizer.step()

**End For**

**End**

---



---

#### Algorithm 3: Build a 3-D Model of the Human Body

---

**Input:** point clouds-src, trg; training times-n.

**Output:** 3-D model of the human body

    Read 3-D model files

    Call algorithm 1 to do.

---

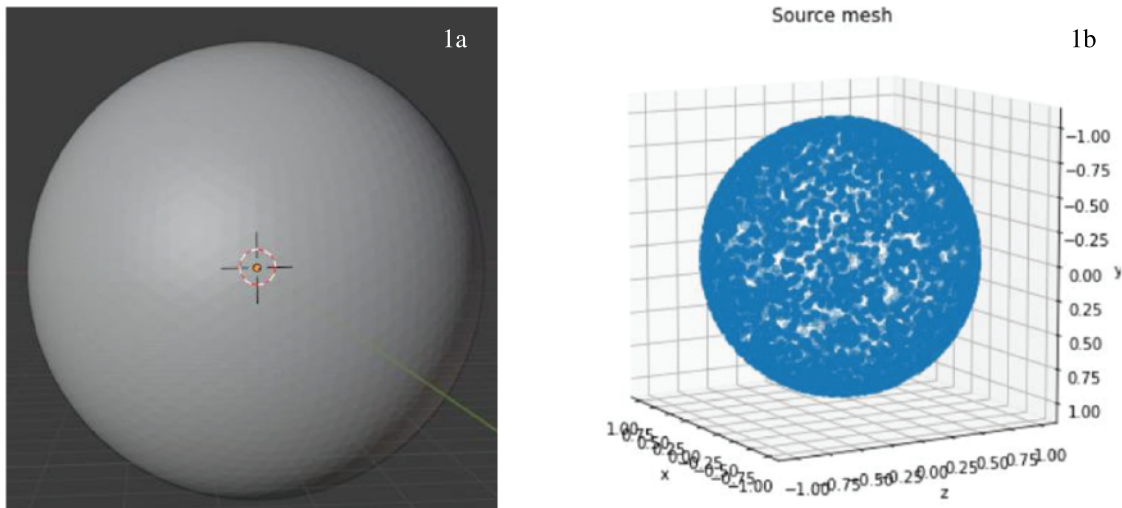
(Continued)

**Algorithm 3:** Continued

Deform 3-D model according to hand measurements by Algorithm 2

**End****3.1 Building the Human Model from Sphere Model**

The input model is first selected as a sphere model make up of 12 fixed points (Fig. 1), after which further points are added, and connecting edges are added to complete the sphere model. From the set of point clouds of the sphere model, we deform this set to fit the set of point clouds of the target model according to the hand measurements.



**Figure 1:** The input model shape (1a) and point set (1b)

After using Algorithm 1, 2, we receive a shape of 3-D human model shows in Fig. 2.

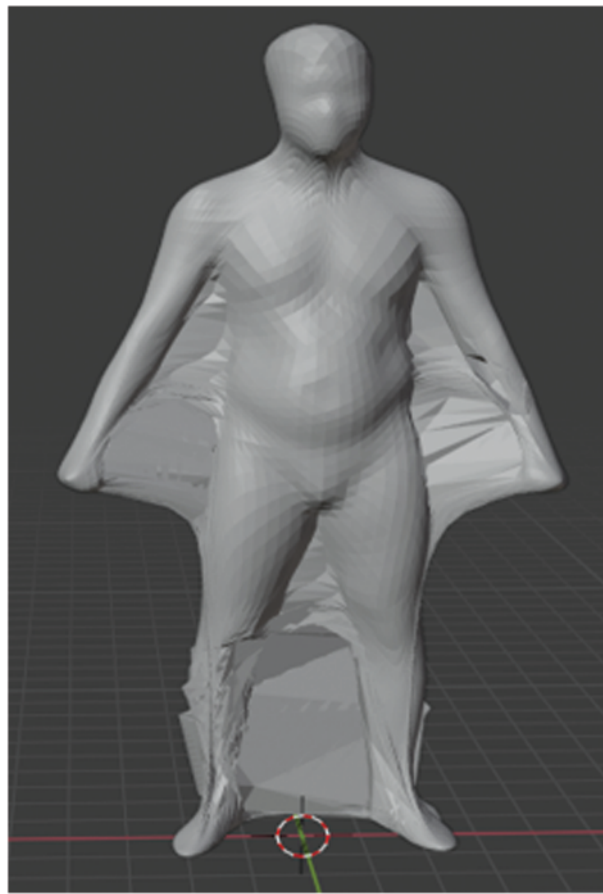
We see that the arms and the legs are sticky together, creating ‘membranes’ between them. We can remove these ‘membranes’, but it takes a lot of time and has a poor result because the shape will appear wrong when removing ‘membranes’.

**3.2 Building the Human Model from 3-D Human Model**

Because the method of 3.1 produces a poor result, we change the input model. Instead of a sphere, we choose an input model of a 3-D standard human model. We choose the so-called standard input model from a set of 3-D human body models that are given. We define the standard model with Eq. (4), shows in Algorithm 4.

$$model = Maxmodels(Max(h_i) - Min(h_i)) \quad (4)$$

where  $h_i$ : height of the  $i^{th}$  model.



**Figure 2:** 3-D human model after training from sphere model

---

**Algorithm 4:** Select the Standard 3-D Human Model

---

**Input:** 3-D model

**Output:** the highest 3-D model

**Max** =  $Max(h_0) - Min(h_0)$ , **Model** =  $Model(0)$

**For** (i = 1 to N):

**If**( $Max(h_i) - Min(h_i) \geq Max$ )

**Max** =  $Max(h_i) - Min(h_i)$ , **Model** =  $model(i)$

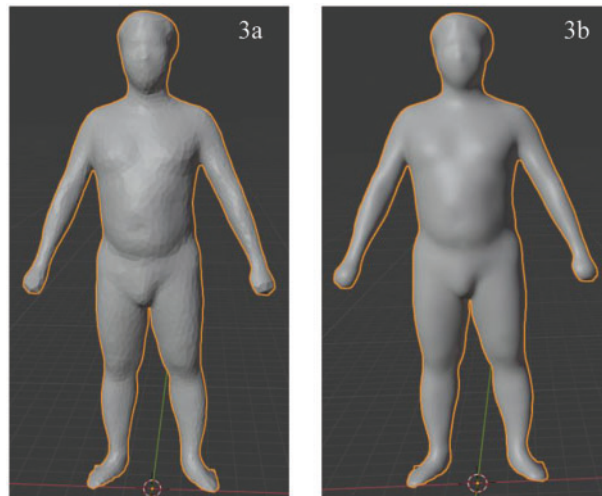
**End if**

**End for.**

**End**

---

The process is identical to changing from a sphere to a person. However, instead of using a spherical model as the input mesh, we use a 3-D model of the human. Fig. 3a shows the results of the test. However, because these models have a rough surface, we decided to add the vn property to the model .obj file to smooth the surface (Algorithm 5), and the result is shown in Fig. 3b.



**Figure 3:** 3-D human model after training from 3-D standard model (3a) and 3-D human model after adding *vn* property (3b)

---

**Algorithm 5:** Adapt 3-D Human Model to Vietnam

---

**Input:** 3-D model

**Output:** 3-D model added *vn* property

**temp** = 0

**nor** = []

**For**  $i = 0$  to  $\text{len}(\text{verts})$

**nor.append**(temp)

**End For**

**End**

---

#### 4 Experimental Results and Discussion

We trained the model using Python and Google Colab platform with graphics processing unit assistance to accelerate the training process. We executed the trained model on the personal computer with an Intel Core i5 7th Gen processor and 8GigaByte random access memory.

The initial data was gathered from two sources, emphasizing youthful and middle-aged persons of both sexes, including over 1000 human models (378 male models and more than 600 female models) provided by the garment research group at Hanoi University of Technology [14]. More than 3000 foreign human models (more than 1500 male models and 1500 female models) were gathered by Yipin Yang et al. [15].

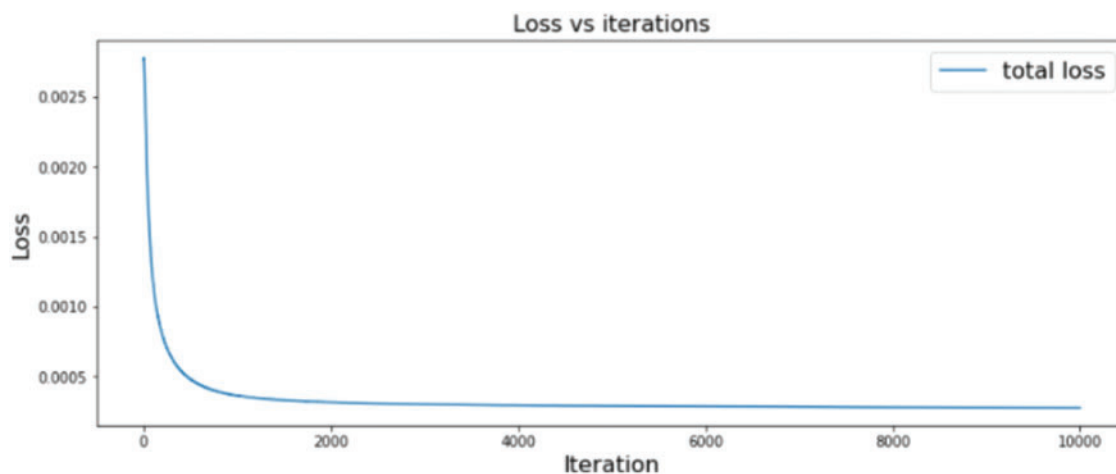
The errors acquired from experimenting with the equipment as mentioned above, with 12500 data points and 5000 data points, are shown in Tab. 1.

From Tab. 1, we can see that the error rate depends on the data points, the more points, the lower the error rate.

**Table 1:** Error of 3-D human models

Error	12500 points	5000 points
Average	0.00026	0.000615
Max	0.000322	0.000675
Min	0.000232	0.000572

After training the 3-D model with 10000 iterations, we found that the error rate significantly decreased in the first 1000 iterations, then slightly reduced. The difference between loops is minimal from the 6000 iterations onwards. Based on Fig. 4, we trained the 3-D model with 7000 iterations to save training time. The error of slices of 3-D models were calculated with the formula in [3], applied to Tabs. 2 and 3.

**Figure 4:** 3-D model error on each loop**Table 2:** Error of principal slices of 3-D human models (cm)

Slices	12500 points	5000 points
Neck	0.4711	1.50532
Chest	0.4966	2.2675
Stomach	0.4689	1.493659
Buttocks	0.5543	1.213283
Upper arm (left)	0.7402	1.45723
Upper arm (right)	0.8505	1.086267
Wrist (left)	0.7366	1.404546
Wrist (right)	0.7608	1.16059
Thigh (left)	0.3234	0.9478933
Thigh (right)	0.4167	0.9461028

(Continued)



**Table 2:** Continued

Slices	12500 points	5000 points
Calf (left)	0.556	1.496387
Calf (right)	0.6215	1.442509
Ankle (left)	0.3653	1.486258
Ankle (right)	0.354	1.167027

**Table 3:** Error of principal slices of 3-D male models and 3-D female models (cm)

Slices	Male	Female
Neck	0.4711	0.39228
Chest	0.4966	1.89187
Stomach	0.4689	0.46008
Buttocks	0.5543	0.73364
Upper arm (left)	0.7402	0.5525
Upper arm (right)	0.8505	0.61004
Wrist (left)	0.7366	0.71498
Wrist (right)	0.7608	0.7865
Thigh (left)	0.3234	0.53612
Thigh (right)	0.4167	0.40858
Calf (left)	0.556	0.43845
Calf (right)	0.6215	0.4083
Ankle (left)	0.3653	0.48125
Ankle (right)	0.354	0.37997

Moreover, as shown in [Tab. 3](#), 3-D male and 3-D female models also have differences in error. Specifically, the chest error of 3-D female models was higher than that of 3-D male models because they differed from the body structure.

The more the number of data points, the more decrease in error, implying that the error rate is proportional to the number of points in the model. More data points lead to more comprehensive and precise results, increasing accuracy and lowering error.

To verify the effectiveness of the method compared with other methods, [Tab. 4](#) shows the results of errors of the method compared with other methods. Our method has errors lower than other methods, including interpolation [3], CNN [4].

The proposed method reduces the computational step since we train on the surface (known as the skin) and the entire model synchronously, without dividing into main slices and separate parts (arms, legs, body, etc.) like interpolation [3] and CNN [4] approach. The interpolation approach [3] is calculated based on the 14 main slices of the garment standard, which leads to the joining of slices leading to differential errors in the joined parts. The CNN approach [4] is based on interpolation from circles to body slices. Although this approach is practical, it reduces the number of computations in slices and increases the number of iterations because each certain distance is calculated once. However, there is also a mistake in concatenating the rings together (like [4]).

**Table 4:** Errors of other methods

Error	Interpolation [3]	CNN [4]		Proposed method			
		Male	Female	Male		Female	
				5000 points	125000 points	5000 points	125000 points
Average	11.02621	0.09860	0.13160	0.00062	0.00027	0.00059	0.00026
Max	125.55837	—	—	0.00068	0.00031	0.00069	0.00032
Min	1.38504	—	—	0.00057	0.00023	0.00059	0.00023

## 5 Conclusion

As a result, in practical applications, particularly in clothing, it is essential to have a reasonably realistic human model in terms of shape and size without needing to be extremely specific about each part and minute detail. Then, in terms of both time and accuracy, building a model with 5000 data points is a viable alternative.

Machine learning, especially deep learning, has been used to develop a method for creating a 3-D model of the human body. This technique takes less time to execute and has fewer errors than prior approaches. However, this technique has several drawbacks, including that the resulting model has a rough surface and needs to add a smoothing step. Besides, the resulting model makes errors about shape, especially the wrist. There are drawbacks that we could not fix in research time. We will develop automatic smoothing algorithms while training 3-D models and improve 3-D model algorithms with more than data points to reduce error in the shape of the wrist in the future.

**Funding Statement:** The authors received Funding for this study from Sai Gon University (Grant No. CSA2021–08).

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] N. Shahrubudin, T. C. Lee and R. Ramlan, “An overview on 3D printing technology: Technological, materials, and applications,” *Procedia Manufacturing*, vol. 35, pp. 1286–1296, 2019.
- [2] J. M. Ai and P. Du, “Discussion on 3D print model and technology,” *Applied Mechanics and Materials*, vol. 543–547, pp. 130–133, 2014.
- [3] T. T. M. Kieu, N. T. Mau, L. Van and P. T. Bao, “Combining 3D interpolation, regression, and body features to build 3D human data for garment: An application to building 3D Vietnamese female data model,” *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 1, pp. 1–9, 2020.
- [4] M. T. Nguyen, T. V. Dang, M. K. T. Thi and P. T. Bao, “Generating point cloud from measurements and shapes based on convolutional neural network: An application for building 3D human model,” *Computational Intelligence and Neuroscience*, vol. 2019, pp. 1–15, 2019.

- [5] Y. Zeng, J. Fu and H. Chao, "3D human body reshaping with anthropometric modeling," in *Int. Conf. on Internet Multimedia Computing and Service*, Springer, Singapore, pp. 96–107, 2017.
- [6] Y. Bengio, A. Courville and P. Vincent, "Representation learning: A review and new perspectives," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.
- [7] R. Scopigno, P. Cignoni, N. Pietroni, M. Callieri and M. Dellepiane, "Digital fabrication techniques for cultural heritage: A survey," *Computer Graphics Forum*, vol. 36, no. 1, pp. 6–21, 2017.
- [8] L. Deng and D. Yu, "Deep learning: Methods and applications," *Foundations and Trends in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014.
- [9] J. Jeong, H. Park, Y. Lee, J. Kang and J. Chun, "Developing parametric design fashion products using 3D printing technology," *Fashion and Textiles*, vol. 8, no. 1, pp. 1–25, 2021.
- [10] B. Yoshua, "Learning deep architectures for AI," *Foundations and Trends Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.
- [11] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [12] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [13] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu *et al.*, "Deep learning for 3D point clouds: A survey," *Institute of Electrical and Electronics Engineers Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 12, pp. 4338–4364, 2021.
- [14] T. T. M. Kieu and S. Park, "Development 'Aodai' pattern for Vietnamese women using 3D scan data," in *Korean Society for Clothing Industry (KSCI) Int. Conf.*, Taiwan, pp. 390–391, 2011.
- [15] Y. Yang, Y. Yu, Y. Zhou, S. Du, J. Davis *et al.*, "Semantic parametric reshaping of human body models," in *2014 2nd Int. Conf. on 3D Vision*, Tokyo, Japan, vol. 2, pp. 41–48, 2014.