Journal of Artificial Intelligence and Data Mining (JAIDM), Vol. 10, No. 1, 2022, 63-74.

Shahrood University of Technology

Journal of Artificial Intelligence and Data Mining (JAIDM)
Journal homepage: http://jad.shahroodut.ac.ir

Research paper

Deformable 3D Shape Matching to Try on Virtual Clothes via Laplacian-Beltrami Descriptor

Hamed Fathi and Alireza Ahmadyfard* and Hossein Khosravi

Faculty of Electrical Engineering, Shahrood University of Technology, Shahrood, Iran.

Article Info

Article History:
Received 22 April 2021
Revised 10 September 2021
Accepted 01 November 2021

DOI:10.22044/JADM.2021.10749.2212

Keywords:
Virtual Clothes, 3D Mapping, Laplace-Beltrami, Shape Deformation.

**Corresponding author: ahmadyfard@shahroodut.ac.ir (A. Ahmadyfard).*

Abstract

Recently, significant attention has been paid to the development of virtual reality systems in several fields such as commerce. Trying on virtual clothes is becoming a solution for the online clothing industry. In this paper, we propose a method for the problem of virtual clothing using 3D point matching of a selected cloth and the customer body. For this purpose, we provide a 3D model of the customer and the selected clothes and put up on the mannequin using a Kinect camera. As the size of the abdominal part of the customer is different from the mannequin, after pre-processing of the two captured point clouds, the 3D point cloud of the selected clothes is deformed in order to fit the 3D point cloud of the customer's body. We use the Laplacian-Beltrami curvature as a descriptor in order to find the abdominal part in the two point clouds. Then the abdominal part of the mannequin is deformed in a 3D space to fit the abdominal part of the customer. Finally, the head and neck of the customer are attached to the mannequin poinIt. The proposed method has two main advantages over the existing methods for virtual clothing. First, no need for an expert to design a 3D model for the customer body and the selected clothes in advanced graphical software such as unity. Secondly, there is no restriction for the style of the selected clothes and their texture, while the existing methods have such restrictions. The result of the experiments shows the ability of the proposed method for virtual clothing in comparison to the existing methods. The visual assessment is the criterion for the evaluation of the virtual clothing methods.

1. Introduction

Virtual reality (VR) is a novel machine vision approach that helps integrate the real and virtual worlds. In this context, a new environment can be experienced virtually even if it is not available in the real world. As an example, virtual clothing can help a customer to try on clothes virtually from a shop. It is not so convenient to try on different clothes in order to find and buy the favorite one. The smart mirror, with the help of virtual reality, is almost capable of alleviating the difficulties. Inadequate space in the changing room, insufficient cloth hanger clamps for the selected clothes, and the need to return to the shopping room for other choices are several difficulties for the customer. As a consequence, many people are not interested in buying the clothes that need to try on. Recently, virtual reality has introduced amazing tools to the fashion world. One of the most outstanding and interesting tools is smart trying on. The smart changing room makes it fun to buy clothes. Using this technology, a customer with the selected piece of clothing can be displayed on a monitor without the requirement to wear it. This technology is equipped with a smart mirror. This mirror is a monitor in which the customer sees his/her photo while wearing the

selected clothes. Also additional options on the mirror for choosing different sizes and colors are feasible.

2. Related Works

Hilsmann et al. [1] have suggested a virtual clothes fitting method using a 2D camera to capture the customer's photo. The customer stands in front of the camera considered to have a green T-shirt with a recognizable logo. In this algorithm, the selected logo is virtually replaced with the existing logo on the clothes worn by the customer.

In this method, the image of the worn T-shirt is segmented, then it extracts the model of the real logo surface (wrinkles) and transforms it on the virtual surface with the help of the mesh vertices' locations on the real logo (customer's cloth). At this point, the optical flow algorithm in [2] and [3] is applied in order to find the mesh vertices on the logo. Zhang et al. [4] have proposed a method to recognize the area and size of the person's clothes using two cameras; on top of the head, and in front of the body. Next, they virtually put a clothing model on its body similar to the worn clothes. Selecting the clothing type is limited according to the notion that one can only select a piece of clothing similar to his/her ones. This algorithm is composed of four parts [4]:

- Recognizing the person's body,
- Recognizing the direction of the person,
- Recognizing the clothing type,
- Mapping the clothing model on the person photo.

In fact, this approach is looking for a match between the dress and the customer by recognizing the clothing and its type. The clothing dataset has been provided using eight men and women [4]. The algorithm imposes some assumptions for the recognition of the person's body. For example, the person must stand where there is a green curtain behind him/her, and a white surface is below it. This approach recognizes the direction of the person to properly show him/her in the virtual mirror. For this purpose, one should drag the person in the image from the top and fit it in an ellipse. Eventually, the orientation of the smallest ellipse diameter shows the person's direction. By introducing the Kinect camera from Microsoft, extracting the point cloud of the objects in front of the camera becomes possible in a 3D space. Accordingly, this camera is utilized to extend an efficient 3D imaging in machine vision. The researchers have also considered using it for virtual clothes by acquiring the 3D model of the customer's body.

Kotan et al [5] have proposed an approach for virtual clothes. Once the customer stands in front of the virtual mirror, s/he can choose a cloth from the user menu. Afterward, the desired cloth appears on a human 3D model, and the method operates in a way that the person can try different sizes. They use the graphic software called the human builder to produce human models. Utilizing this software, one can create a model by applying different sizes and skin colors. Afterward, the point cloud and the color image of the customer taken by the Kinect camera are given to the graphic software (Unity) in order to estimate the size of the person. Then the corresponding model of the customer's body selected in the graphic software is shown on the monitor. In the next step, the customer can choose the desired cloth to see it on the created model. The 3D model follows the person's movements by the connection of the Unity software and Kinect camera. As it can be realized, in this method, a skilled graphic designer is required to model the clothes in Unity. Therefore, this method is not suitable for unskilled users. Yang et al. have followed three different scenarios for virtual clothing [6]:

- Virtual clothing on the model or a mannequin,
- Virtual clothing on the customer's photo,
- Virtual clothing on the 3D model with the face of the customer.

In the first scenario, the created virtual dress is placed on the person's 3D model, and with the help of a 3D to 2D mapping, the monitor (virtual mirror) shows the image in which the person has worn a piece of clothing. In the second scenario, the clothing is mapped on the customer's photo by a 3D to 2D mapping. The third scenario is similar to the first one but the face of the customer appears on the monitor instead of the face of the model (avatar).

In the addressed research, the algorithm in [7] is adopted to create a virtual dress. This method exploits 3D datasets (CAESAR) instead of creating a 3D model for the customer's body. These datasets consist of the human point clouds for men and women with different styles, fat, thin, tall, and short. Then the selected model is modified using 20 extracted measurements from the 3D customer's body. The measurements include the person's height, shoulder width, waist, hip, chest, thigh, ankle circumference, waist, knee, jaw, arm height, and forearm length. In fact, for each person, a similar avatar is selected automatically from the database. Next, the model is modified to better match the person's size.

In this approach, in order to set the skin color of the model to the customer's skin, first, the face region is identified with the ACM algorithm [8]. Then the skin color of the avatar is set to the customer's skin using the technique suggested in [9].

After obtaining the virtual dress and the 3D model of the customer, in the first scenario, this method uses the Unity software to provide a virtual dress. In the second scenario, in order to map the dress on the person's body, a 3D to 2D mapping is used [10].

For this purpose, in the j th frame of the video, the mapping matrix P_j transforms every point in a 3D space to the corresponding point in a 2D space. The optimal P_j is obtained by minimizing the sum of the squares of the distance between the 3D and 2D points:

$$P_j = \underset{P_j}{argmin}\{\omega_J E_J + \omega_B E_B + \lambda \left\| P_j \right\|^2\} \quad (1)$$

where E_j and E_B are defined in (2) and (3):

$$\mathrm{E_J} = \sum_{\substack{\mathrm{M_i} \in \mathrm{J_{Avatar}} \\ , \ \mathrm{m_i} \in \mathrm{J_{User}}}} \left\| \mathrm{m_i} - \zeta\left(\mathrm{P_j}, \mathrm{M_i}\right) \right\|^2 \quad (2)$$

$$\mathrm{E_B} = \sum_{\substack{\mathrm{M_i} \in \mathrm{B_{Avatar}} \\ , \ \mathrm{m_i} \in \mathrm{B_{User}}}} \mathrm{m_i} - \zeta\left(\mathrm{P_j}, \mathrm{M_i}\right)^2 \quad (3)$$

$\zeta\left(\mathrm{P_j}, \mathrm{M_i}\right)$ is a mapping from point $\mathrm{M_i}$ using transformation $\mathrm{P_j}$ in the j th frame of the video $\mathrm{J_{Avatar}}$ and $\mathrm{B_{Avatar}}$ are skeletal point set obtained by Kinect, and the set of points on the skeleton acquired from the 2D picture of the person, respectively. The control parameter λ is set to 0.0002. One can accomplish 3D to 2D mapping by minimization of (3) using the LM algorithm [11]. The available 3D approaches utilize graphic software such as Unity to generate the cloth model on the customer's avatar [5]. A skilled graphic designer is required to build a 3D clothing model on the avatar. Moreover, generating clothes with complex textures in a graphic software is very complicated. These two problems restrict the use of graphic software for virtual clothing, where an avatar model is created.

In [12], the author has categorized the size of clothes with 3D measurements such as the circumferences of chest, waist, hip, thigh, and knee by Kinect camera. Using the Unity3D game engine overlays 3D clothing virtually on the user. Article [13] has tried to provide a virtual dressing and follow most physical movements that people can use at home. In this work, an avatar model is created with the help of the customer's face image and hairstyle so that the customer has a more real experience. They also use Unity3D to cover the clothes.

Many works have been conducted to develop a method with a more realistic result. The problem with the methods reviewed previously is that a model for each piece of clothes must be constructed by a graphic designer. The design of a 3D model for clothing is a challenging task. In [14], a three dimensional clothing model has been constructed using 2D images. Pons-Moll et. al. [15] have used high-resolution scanners (60 frames per second) to create a 3D model of clothes. In [16], a new approach for creating a 3D model from a set of clothing images has been proposed. Unlike the previous methods that require the participation of humans or mannequins, the authors of [16] have claimed that they make clothes from only two images of the front and back of the 3D model using a JFNET learning network.

In all methods that are seeking to make a 3D model of a dress without the intervention of a graphic designer, the output results have an animated mode like [14] and [15] or unnatural like [16], which estimates the 3D model of a dress from a photo. They are all far from reality. The mentioned methods always require a graphic designer to design clothes using graphic software and animation production. Therefore, these restrictions prevent the clothing industry from readily using a virtual try-on application.

More recently, Li et al. [17] have proposed a virtual dressing method organized in two steps: reconstructing a 3D model for customer and virtually putting a selected clothes on the model. First, using a frontal and side image of the customer, a 3D model of the customer body from the CAESAR dataset is selected. Next, the image of the person is mapped on this model. This algorithm uses 2D to 3D mapping in order to provide a 3D model of the customer. Then the image taken from the piece of clothes is mapped on the 3D model of the customer. The 3D model of the person, provided using its 2D images, is far from reality.

3. Proposed Method

In this article, it is assumed that the size of the abdominal part of the customer body is different from that of the mannequin. Hence, we attempt to adapt the size of the virtual clothes on the

mannequin body to the size of the customer's abdominal.

In this approach, we perform the virtual trying-on without modeling the clothes or customer in the graphical software. Instead, we collect the 3D point clouds as well as the 2D color image from both the customer body and the mannequin worn the desired clothes for virtual trying on using a Kinect camera. In our experimental set up, a customer is placed on a turning plate in front of the Kinect camera. Thus the 3D point cloud of his/her body is captured. At the same time, the color image of the customer is taken by Kinect. In the same way, the point cloud and color image of the clothes put on the mannequin are captured as the extracted point clouds are contaminated with the outliers and noise that cause the error. In a pre-processing step, the outliers are filtered. After this step, two filtered point clouds from raw data are extracted.

After preparing the point cloud, the cloud surfaces are described using the special Laplace-Beltrami eigenvectors. Finally, K-means clustering is used to segment the abdominal part of the customer and mannequin body. Then an acceptable output is obtained by the size of the mannequin's abdomen and replacing the mannequin's head area with the customer's head. The diagram of the proposed method is shown in Figure 1.

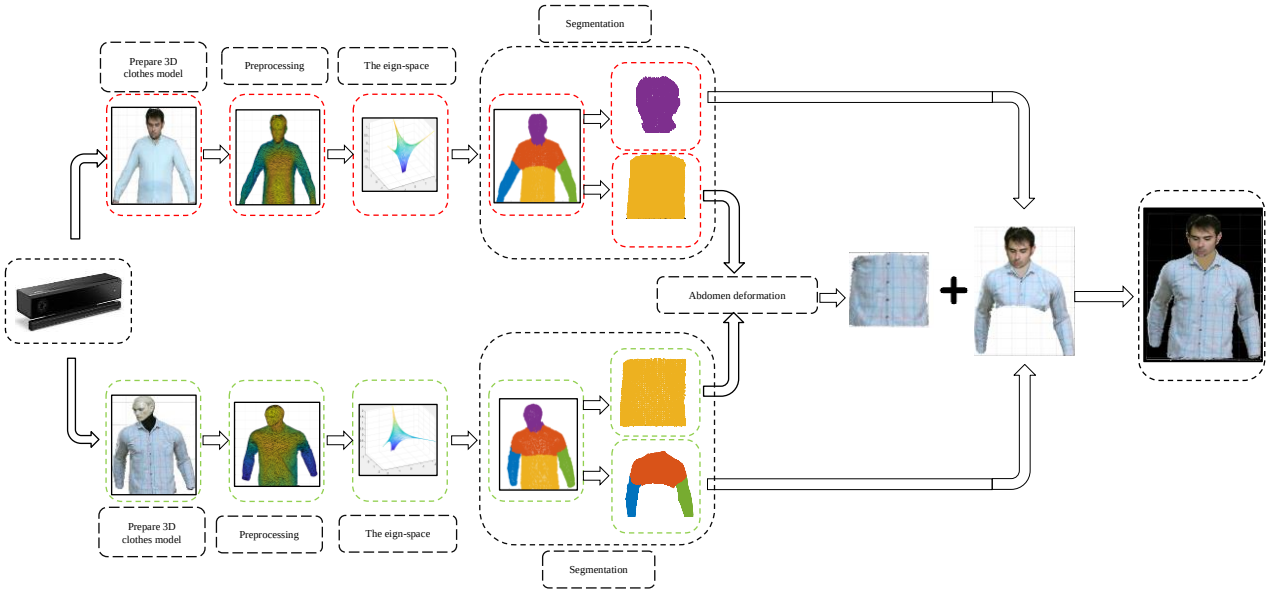


Figure 1. Diagram of the proposed method.

3.1. Point Cloud

In the first step, the 3D point clouds of the desired piece of clothes and the customer for virtual trying on are captured using the Kinect camera. For this purpose, a turning plate was designed to turn the mannequin while wearing the desired clothes. In front of the turning plate, the Kinect was set on the tripod. The speed of the turning plate is set so that the 3D scan can accurately extract both the color and the 3D point cloud.

Similarly, when a customer goes shopping to buy a cloth, s/he is asked to stand on top of the turning plate. Using a Kinect camera in the front of the customer, the 3D point cloud and color image of the trying-on person are captured (Figure 2).

The captured 3D data from the model and trying-on person is required to be pre-processed to remove the clutter and noise from the point clouds.

(a) (b)

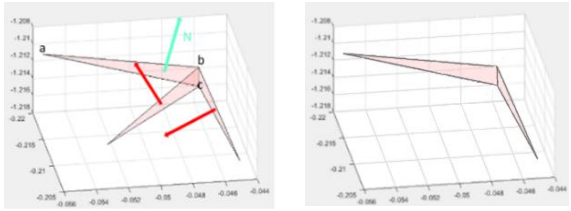
Figure 2. (a) 3D point cloud of a selected clothes on the mannequin (b) a 3D point cloud of the customer.

3.2. Pre-processing

The captured point cloud is contaminated with clutter and noise. Here, there is a problem; for some 3D positions, more than one data is recorded. The duplicated data in the point cloud causes the problem while constructing the surface

mesh. In the first step, the duplicated data is removed from the point cloud. Then using the PCL library, a surface mesh for the filtered point cloud is constructed. We developed a mex function in Matlab to use the mesh construction function in the PCL library, developed in C++. The mesh obtained from the PCL library has some redundant parts that make errors in the subsequent processing, and thus that are required to be removed. This process includes the following steps: (1) removing the points that are out of the surface mesh, (2) removing the redundant edges, and (3) removing the redundant meshes.

The points that do not lay on the main surface mesh must be removed from the point clouds. The presence of these points provides the edges that are common between more than two triangles (Figure 3).



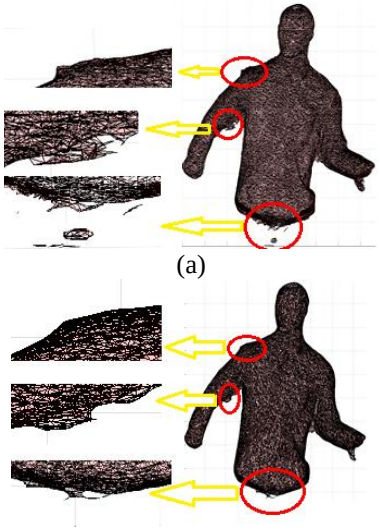
(a) (b)

Figure 3. (a) An example in which the edge, bc, connects more than two triangles (b) after removing the undesired triangle on the mesh.

Accordingly, these vertices from the mesh are removed, and the mesh is reconstructed afterward. In a well-defined mesh, each edge joins at most two triangles in the mesh. However, as mentioned earlier, some edges in the mesh join more than two triangles (Figure 3-a). After detecting such an edge, the redundant triangle connected to the edge is removed. The triangles constructing a smooth surface are kept, and the undesired triangles are removed. We determined the angle between the normal vectors corresponding to the triangles join at one edge. We kept two neighboring triangles if the angle between their normal vectors were small. The normal vector of each triangle was calculated by the cross-product of pair of its edges (the common edge and the other one).

Note that regarding the removal of the redundant triangles from the mesh surface, the mesh vertices and the point cloud are modified.

In other words, the third unshared vertex of the removed triangle must be removed from the point cloud matrix and two vertices of the common edge remain. One more issue is required to be considered on the mesh surface. The mesh surface consists of a number of sub-meshes. The main sub-mesh presents the body, and the remaining sub-meshes are related to the clothes, inaccuracy in capturing procedure or the result of mesh modification in the previous steps. In this respect, the sub-meshes of the cloud are searched and the sub-mesh with the maximum number of triangles is preserved as the main mesh of the point cloud. Figure 4 depicts the elimination of redundant sub-meshes and smoothing of the surface mesh.



(a)

(b)

Figure 4. Surface mesh (a) before pre-processing (b) after pre-processing.

The obtained mesh from the point cloud of the 3D model has an uneven surface. This causes the curvature of the surface to be estimated unreliably. Thus it is necessary to smooth the mesh surface. We adopted the mean normalized curvature as a smoothing filter [18]. The smoothing operation at the i th point of the cloud is formulated as:

$$X^n = \left(I - \lambda \overline{K_n}\right) X^{n+1} \qquad (4)$$

where X^n stands for the matrix $(N*3)$ of mesh vertices at the n th iteration of the smoothing process. n represents the number of smoothing stages, $\overline{K_n}$ is the mean normalized curvature, and λ is the smoothing factor. The mean normalized curvature is determined as:

$$(\bar{K}_n)_{norm} = \frac{1}{\sum_j \left(\cot\left(\alpha_j^l\right) + \cot\left(\alpha_j^r\right)\right)} \times \sum_j \left(\cot\left(\alpha_j^l\right) + cot\left(\alpha_j^r\right)\right)\left(X_i - X_j\right) \qquad (5)$$

where α_j^r and α_j^l are the right and the left angles of the common edge e_{ij} as depicted in Figure 5.

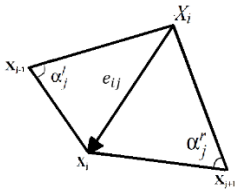
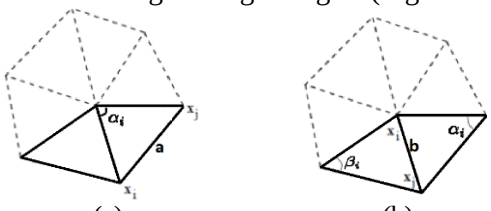


Figure 5. Diagram of two neighboring triangles on a surface mesh [18].

3.3. Laplace-Beltrami

Different criteria can describe the curviness of each point on a surface in 3D space: main curvatures, Gaussian curvature, and mean curvature. The mean curvature normal operator [19] is known as the Laplace Beltrami operator. Based on the conventional method in the field of geometric modeling of meshes, the Laplace-Beltrami matrix is required to be first obtained to calculate the mean curvature [19]. There are two types of edges in a mesh, the edges located in the mesh boundary (Figure 6-a) and those common between two neighboring triangles (Figure 6-b).



(a) (b)

Figure 6. (a) Border edge (type A) and (b) Joint edge [19].

The Laplace-Beltrami matrix (L) is an $n \times n$ square matrix (n is the total number of mesh apexes) that specifies the relation between every two vertices connected to an edge in the cloud. This weighted relationship is obtained based on the cotangent of the angle or angles across the edge. L_{ij} the element at the i th row and the j th column of the Laplace-Beltrami matrix are determined by (6), [19] :

$$L_{ij} = \begin{cases} -w_{ik} & i \neq j \\ \sum_{k} w_{ik} & i = j \end{cases} \quad (6)$$

where w_{ij} is the weight associated to the angles in front of the $\overline{X_i X_j}$ edge. It is worth noting that the Laplace-Beltrami matrix is a sparse matrix since the number of vertices adjacent to the i th edge of the mesh is small. If $\overline{X_i X_j}$ is a boundary edge (Figure 6-a) (7), is used to obtain the weight w_{ij} between the i th and j th vertices.

$$w_{ij} = \frac{1}{2}\cot(\alpha_i) \quad (7)$$

where, α_i is the angle in front of the edge $\overline{X_i X_j}$; otherwise, for the interior edges,(8) , is used.

$$w_{ij} = \frac{1}{2}\cot(\alpha_i + \beta_i) \quad (8)$$

where α_i and β_i stand for the two angles on either side of the common edge of two neighbouring triangles, respectively, as shown in figure 6-b [19]. As it can be seen from Figure 6, a point with high curvature means that α_i and β_i increase considerably. The variation range of these angles is from 0 to 90 degrees. Hence, the weights corresponding to the angles decrease by increasing the curvature at each point, and vice versa. The distribution of the neighbouring weights can be considered for each point in the Laplacian matrix, L, whereby the curvature is determined.

4. Virtual Dressing

We propose a virtual dressing method different from the previous approaches. In this method, we aim to fit the selected clothes worn on the mannequin to a customer's 3D mesh, and provide a 2D color image of the customer on a virtual mirror. The previous methods have some drawbacks such as time complexity, insufficient accuracy, and low quality of the final result. For mapping the clothes of the mannequin to the customer's abdomen, a time-consuming process is required to search all the mannequin points and the abdomen to find the corresponding points. On the other hand, the search time will be reduced by decreasing the number of points in the two clouds but the result on the virtual mirror would have an unacceptable resolution.

For this reason, we propose to modify the size of the clothes on the mannequin based on the size of the customer. In this method, the point cloud of the head of the customer is placed on the right position as the head of the mannequin model. It is assumed that the size of the customer's abdomen is the major parameter for fitting the customer's body to the mannequin. Segmentation is an idea that can help us to find model's abdomen. But 2D segmentation [20], cannot be used. We should use 3D segmentation to segment 3D objects. Thus using the proposed segmentation on the point clouds of the customer and mannequin, we find the abdominal area for two point clouds. Then the shape of the abdomen area for the mannequin model is modified to fit the abdominal area of the

customer. For this purpose, it is necessary to segment two point clouds into the corresponding parts. Next, we focus on the problem of meaningful point cloud segmentation that enables the proposed method to establish the correspondence between the customer and mannequin parts. In [21], Kleiman et al. have presented a method based on the curvature characteristics of the point cloud surface to segment the point cloud in 3D space. A proper result has been reported [21] but the accuracy of this method for segmentation to corresponding parts depends on the similarity of the two data and also the number of points in the two clouds [21]. Huang et. al. [22] have addressed the problem of multi-body segmentation and motion estimation. The aim is to register rigid 3D shapes using multiple scans at the same time, which are represented as point clouds, and the objects are from the unseen categories. In [23], a semi-supervised approach called Robust Learning of One-Shot 3D Shape Segmentation (ROSS) has been proposed. The method uses labeled shapes for training. In [24], a point cloud has been represented from adaptive co-segmentation of a set of 3D shapes of a deep neural network. George et al. [25] have proposed a deep learning approach for 3D segmentation. The implementations of several deep learning techniques such as neural networks (NNs), auto-encoders (AEs), and CNNs have been reported [25]. The method in [26] is an efficient approach to compute joint graph layouts and use it for visualizing segmented meshes. Poulenard et al. [27] have proposed an approach for performing signal convolution on curved surfaces, and the diversity of geometric deep learning applications shows its usefulness. In [28], an architecture called Synchronized Spectral CNN (SyncSpecCNN) has been introduced. Xu et al. [29] have presented a method based on the ideas that can motivate future research in analyzing and processing data-driven shape. In [30], a deep architecture has been presented for 3D object segmentation into their labeled semantic parts. In order to obtain a method for multi-scale isometry invariant mesh segmentation in [31], it combines the persistence-based clustering with the Heat Kernel Signature (HKS) function.

In this work, we used a segmentation method similar to [32]. Laplace-Beltrami eigenvectors and k-means clustering are used for segmentation due to their lower complexity in contrast to the other reviewed methods [22-31]. Each one of the Laplace Beltrami's eigenvectors represents a specific type of curvature that can be used to segment the point cloud. It should be noted that the first vectors represent low curvatures, and the higher the vectors, the more details of the curvature of the points cloud. By examining the eigenvectors of the Laplace- Beltrami matrix from the 3D point cloud model, it can be deduced that there is a feature for each point to the number of eigenvectors. In other words, we will have a feature of curvature by having an eigenvector matrix of Laplace-Beltrami matrix for any point with the dimensions equal to the number of feature vectors. From this viewpoint, we reached the next M space (M equal to the number of vectors) from 3D space. On the other hand, we do not need the feature of partial curvatures, and we will examine the features of curvatures for segmentation in general. For this purpose, we introduce the vectors 2, 3, and 4 as new dimensions for points [33]. The first vector is also not considered since it does not have useful information about curvature. By doing this and drawing points in the new 3D space, we find that the shape of the point cloud becomes the same in the new feature space of any type of cloud point data, as shown in Figure 7.

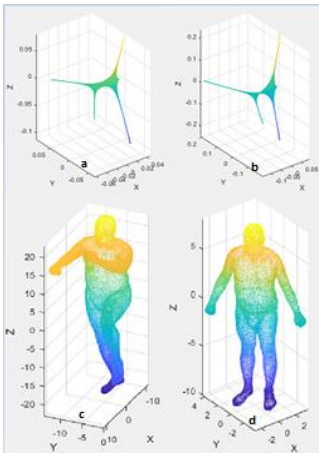


Figure 7. Representation of the point clouds in the eigen-space (a) and (b) and the corresponding point clouds (c) and (d).

By perusing the shape of the point cloud in the new space, we will conclude that the shape of the points cloud is a kind of skeleton of the data in the feature space form of the points cloud in the original 3D space. In Figure 7-a and 7-b, it is derived that a corresponding segmentation can be achieved in the original 3D space by corresponding segmentation of two data of point cloud in the feature space. Notably, the shape of the cloud points in the feature space is independent from the rigidity and non-rigidity of the point cloud data, and the shapes are similar for both the rigid and non-rigid modes of the feature space. Figure 8 shows that we obtain a corresponding segmentation between the points

cloud in the original 3D space by clustering the shape of the feature space.

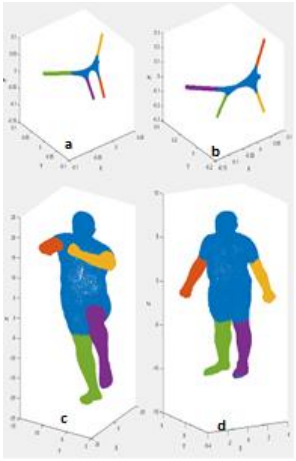


Figure 8. Segmentation of a pair of point clouds using clustering in the eign-space (a) and (b) representation of the point clouds (c) and (d) in the eign-space.

On the other hand, the similarity of two point clouds in the feature space and also the similar clustering of points cloud in this space can be used to achieve a corresponding segmentation between the two point clouds in the main 3D space. For this purpose, the K-means algorithm is used to cluster the points in the feature space [32]. Next, we use the indices of each point in the feature space and segment the points cloud in the main space. Then the sections are required to be sorted and the indices are required to be matched in both data. If the clustering algorithm is run on one data twice, the result may be different each time so that performing synchronization is required. For this purpose, the indices are required to be arranged according to the location of each section, and a new index should be assigned to the relevant section. In this work, the position of the representative points in the third axis of the main 3D space is used to sort each section. The representative points of each section are the points of that section with the least value in the third axis. Figure 9 shows the results of the clustering algorithm between two non-rigid data. Using these three feature vectors allows the scanner-generated shapes to be segmented as well. In this work, two data from humans and mannequins is prepared using the Kinect version 2 device. In Figure 9, the results of the implementation of the proposed algorithm are given on real data. As it can be seen, the two corresponding 3D models are segmented.

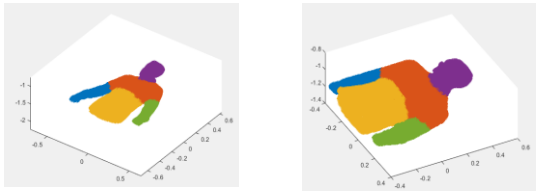
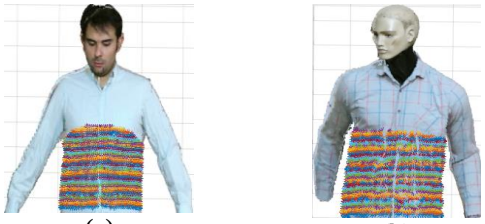


Figure 9. Result of segmentation using the K-means clustering on real world data captured by Kinect camera.

4.1. Resizing mannequin abdomen

Now, suppose that two 3D models of Figure 9 from the mannequin and the customer are prepared and the algorithm is implemented for mapping the corresponding parts between two models. Then we can change the shape of the 3D model of the mannequin by having the abdomen of each model. For this purpose, the abdomen is divided into an equal number of strips in both 3D models of the mannequin and the customer in the direction of the vertical axis. Here, it is divided into 40 segments (Figure 10). The number of segments were selected empirically. A low number of segments would result a gap between the resized parts, and a high number increases the matching complexity.

Next, we will have two almost oval shapes by selecting both corresponding strips of the two models, as shown in Figure 11-a and 11-b.



(a) (b)

Figure 10. Strips in both 3D models of the mannequin and the customer.

The ratio of small to large diameters between the two corresponding strips is the idea of our proposed method. For this purpose, we first create the ellipse. As it can be seen in Figure 11-c and 11-d, large and small diameter of the ellipse are of length and width of a rectangle that can accommodate the desired strip. Next, we will have two almost oval shapes by selecting both corresponding strips of the two models as shown in Figure 11-a and 11-b. The ratio of small to large diameters between the two corresponding strips is the idea of our proposed method. For this purpose, we first create the ellipse. As it can be seen in Figure 11-c and 11-d, large and small diameter of the ellipse are of length and width of a rectangle that can accommodate the desired strip.

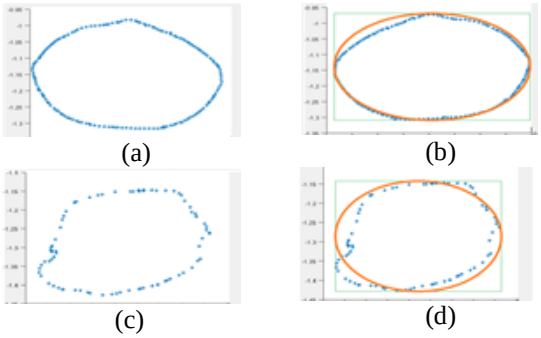


Figure 11. (a) One strip among strips from the customer 3D model (Figure 10) (b) ellips fitted to the strip in (a), (c) one strip from the mannequin, and (d) ellips fitted to the strip in (c).

Now we can use the small diameter ratio (a) and also the large diameter ratio (b) between the two models. The parameters Z and X are multiplied by a and b of the mannequin abdomen point clouds respectively. As a result, we are able to change the dimensions of the mannequin's abdomen area in proportion to the dimensions of the customer's. As it is obvious, the neck area of mannequins has a certain color similar to human skin but the skin color of different people is different. If the neck area of the mannequin is to have its own color, it creates a contradiction for the customer that makes the person not feel comfortable. For this purpose, the neck area of the mannequin is painted black so that this distinction can be used to choose the skin color of the customer and showed in Figure 10-b, which is the 3D model of the mannequin.

The color of the nose tip is used as the neck color to choose the right color. Next, we should separate the areas of the mannequin head and the customer, which is possible using the result of the algorithm for mapping the appropriate parts, presented in this article. Figure 12 shows the separated head areas of the customer using the proposed algorithm.

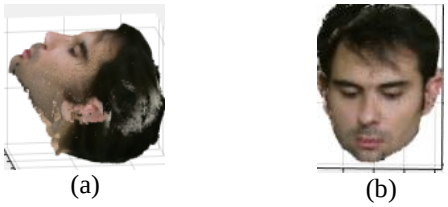


Figure 12. Head area of a customer from two viewing angles.

Afterward, transferring the separated area of the customer's head to the mannequin's body must be done. In order to achieve this, the strip center of the head-to-neck area of both models is used. $O1$ is the center of the head-to-neck area of the customer, and $O2$ is the center of the head-to-neck area of the mannequin. Then we can transfer the head area of the customer to the mannequin body through (9).

$$X = X_m - (O1 - O2) \tag{9}$$

X_m is the coordinates of the points on the customer's head. Figure 13 shows the final result of the proposed virtual dressing algorithm. In order to evaluate the proposed method, the 3D body of two volunteers as customers were captured using the Kinect camera. The 3D model of two shirts was also captured in the same way. Each one of the two shirts were virtually put on either of two customers. The results of the proposed virtual dressing method on the customers are shown in Figure 13. As it can be seen, the proposed method is able to make virtual dressing usable without the need of an expert in the use of graphical software for general designing. The proposed method also considers the discussion of time in addition to preserving the feeling of real dressing. The execution time for this algorithm takes about 40 seconds.

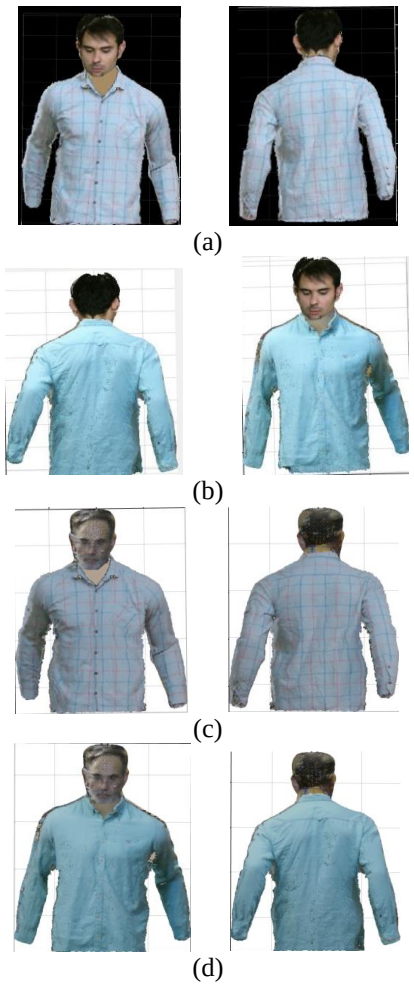


Figure 13. Result of the proposed method for virtual dressing of two customers with two different shirts. Customer1 with (a) the first clothes (b) the second clothes. Customer2 with (c) the first clothes (d) the second clothes.

5. Evaluation

The most proper criterion to evaluate the proposed method is to assess visually the quality of the result for the virtual clothing. In this section, two types of visual criterion are used. We first compare the result of the proposed method with the results of the methods in [13] and [17]. Next the result of the proposed virtual trying on method is compared against the realistic trying on.

By visual comparing of the proposed algorithm results with the methods suggested in [13], [17] (Figure 14), it can be seen that the proposed method has the following advantages. In comparison to [13], the result of the proposed algorithm is closer to reality. In [17], the 3D model of a customer is selected from a database according to the size of the customer. Then the pattern of the desired clothes is painted on the 3D avatar of the customer. In contrast, we directly provide a 3D model of the clothes and map it to the 3D point cloud of the customer body. In [17], the input images should have the same posture of the generic model to the reconstruct human shape. Large occlusion or great posture variation requires a much larger set for training of the ASM model. The presumption of algorithm [17] is having two types of clothing, either like a T-shirt that sticks to the body or like a skirt that fits loosely on the body. However, in fact, there is some gap between body and shirts and even T-shirts. The assumption of [17] necessitates that the output model becomes as if the person's body is painted, and this is far from reality.

In the algorithm [13], the unity software is used to design the 3D model of the customer. In contrast, the proposed method directly constructs the 3D model of the customer and clothes using Kinect easily. Algorithm [17] has not commented in terms of time.

(a) (b)

Figure 14. Result of virtual clothing using (a). the method in [17] and (b) the method in [13].

Similar to [17], a further evaluation was made by comparing the result of virtual trying on method with the customer actually wearing the selected clothes.

Figure 15 shows the result of the proposed method applying for virtual trying on clothes besides the image of customer when really wearing the same clothes. As it can be seen, the quality of the virtual trying on system using the proposed method is comparable to the image of customer wearing the same shirt.

(a) (b)

Figure 15. (a) Image of a person wearing a shirt and (b) result of virtual clothing using the proposed method.

6. Conclusion

The methods proposed for virtual dressing fall into two categories: 2D mapping and 3D mapping. The result of the virtual clothing methods based on 2D mapping is far from reality, and the customer would not be satisfied. On the other hand, the existing 3D mapping methods depend on an expert in the professional graphic softwares in order to construct a 3D model for the customers and clothes. This constraint restricts the applicability of these methods. Among the 3D mapping methods, some methods apply point-to-point mapping in a 3D space, which are both time-consuming and without a sufficient accuracy.

In this paper, we proposed a method for virtual clothing in order to overcome the limitations of the existing methods. The proposed method does not require an expert to construct 3D models in the graphic environments such as Unity.

The proposed method is based on 3D mapping of the clothes on the customer body but its complexity is less than the existing methods. It is practical, and simply provides a 3D model of the customer and the selected clothes.

In the proposed method, first, the point cloud of the selected clothes puts on the mannequin, and the customer body is captured using the Kinect camera. After pre-processing of two points clouds, the surface meshes of these clouds are constructed. These meshes consist of the triangles that define the surface of each points cloud. Due to the complexity of extracting the mesh from the points cloud, with several steps of points processing, we get a smooth flat mesh for each cloud. We describe the surface meshes using the Laplace-Beltrami eigenvectors. These descriptors

are obtained using the normal vector at the surface vertices.

Then by using the cloud segmentation algorithm, the points of the clothing model and the individual are determined with the method based on the Laplace-Beltrami function of the abdomen in both models. After identifying the abdomen of the two models with the algorithm proposed in this article, the abdomen of the 3D model of the mannequin changes according to the abdomen of the customer. After changing the abdominal area of the mannequin model, it is time to transfer the mannequin head area to the neck of the mannequin model instead of the mannequin head area. Finally, we achieve the 3D model of the customer who sees the selected clothes on his body.

References

[1] A Hilsmann, P Eisert, "Tracking and Retexturing Cloth for Real-Time Virtual Clothing Applications," *Computer Vision/Computer Graphics Collaboration Techniques*, pp. 94-105, 2009.

[2] BKP Horn, BG Schunck, "Determining optical flow," *Artificial Intelligence*, Vol. 17, No. 1-3, pp. 185-203, 1981.

[3] P. E. A Hilsmann, "Deformable Object Tracking using Optical Flow Constraints," *Visual Media Production CVMP*, 2007.

[4] W Zhang, T Matsumoto, J Liu, and M Chu,"An Intelligent Fitting Room using Multi-Camera Perception," *International Conference on Intelligent User Interfaces*, pp. 60-69, 2008.

[5] M Kotan, C Öz, "Virtual Dressing Room Application with Virtual Human using Kinect Sensor," *Journal of Mechanics Engineering and Automation*, pp. 322-326, 2015.

[6] M Yuan, IR Khan, F Farbiz, and S Yao, "A Mixed Reality Virtual Clothes Try-On System," *IEEE Transactions on Multimedia*, Vol. 15, No. 18, pp. 1958 - 1968, 2013.

[7] P Volino, N Magnenat Thalmann, and F Faure, "A simple approach to non-linear tensile stiffness for accurate cloth simulation," *ACM Transactions on Graphics*, Vol. 28, No. 4, 2009.

[8] S Milborrow, F Nicolls,"Locating facial features with an extended active shape model," *European Conference on Computer Vision*, pp. 504-513, 2008.

[9] E Reinhard, M Adhikhmin, and B Gooch, "Color Transfer between Images," *IEEE Computer Graphics and Applications*, Vol. 21, No. 5, 2001.

[10] M Özuysal, V Lepetit, F Fleuret, and P Fua,"Feature Harvesting for Tracking-by-Detection," *European Conference on Computer Vision*, pp. 592-605, 2006.

[11] DW Marquardt, "An Algorithm for Least-Squares Estimation of Non-linear Parameters," *Journal of the Society for Industrial and Applied Mathematics*, Vol. 11, No. 2, p. 431–441, 1963.

[12] SB Adikari, NC Ganegoda, RGN Meegama, and IL Wanniarachchi, "Applicability of a Single Depth Sensor in Real-Time 3D Clothes Simulation: Augmented Reality Virtual Dressing Room Using Kinect Sensor," *Advances in Human-Computer Interaction*, Vol. 2020, 2020.

[13] KW Mok, CT Wong, SK Choi, and LM Zhang, "Design and Development of Virtual Dressing Room System Based on Kinect," *I.J. Information Technology and Computer Science*, pp. 39-46, 2018.

[14] S Yang, ZH Pan, T Amert, KE Wang, L Yu, and T Berg, "Physics-Inspired Garment Recovery from a Single-View Image," *ACM Transactions on Graphics*, Vol. 37, No. 5, 2018.

[15] G Pons-Moll, S Pujades, S Hu, and MJ Black, "ClothCap: Seamless 4D Clothing Capture and Retargeting," *ACM Transactions on Graphics*, Vol. 36, No. 4, 2017.

[16] Y Xu, S Yang, W Sun, L Tan, K Li, and H Zhou, "3D Virtual Garment Modeling from RGB Images," *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2019.

[17] C Li, F Cohen, "In-home application (App) for 3D virtual garment fitting dressing room," *Springer Multimedia Tools and Applications*, 2020.

[18] M Desbrun, M Meyer, P Schröder, and AH Barr,"implicit fairing of irregular meshes using diffusion and curvature flow," in *Profceedings of the 26th Annual Conference on Computer Graphics and Interactive Ttechniques*, pp. 317-324, 1999.

[19] X David Gu, R Guo, F Luo, and W Zeng, "Discrete Laplace-Beltrami Operator Determines Discrete Riemannian Metric," *ArXiv*, Vol. 1010.4070, 2010.

[20] S. Bayatpour Seyed M. H. Hasheminejad, "Object Segmentation using Local Histograms, Invasive Weed Optimization Algorithm and Texture Analysis," *Journal of Artificial Intelligence and Data Mining (JAIDM)*, Vol. 9, No. 4, 2021.

[21] Y Kleiman, M Ovsjanikov, "Robust Structure-Based Shape Correspondence," *Computer Graphics Forum*, Vol. 38, pp. 7-20, 2019.

[22] H Wang, T Birdal, M Sung, F Arrigoni,SM Hu, and L Guibas, "MultiBodySync: Multi-Body Segmentation and Motion Estimation via 3D Scan Synchronization," *Computer Vision and Pattern Recognition*, 2021.

[23] S Yuan, Y Fang,"ROSS: Robust Learning of One-shot 3D Shape Segmentation," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[24] C Zhu, K Xu, S Chaudhuri, L Yi, LJ Guibas, and H Zhang,"AdaCoSeg: Adaptive Shape Co-Segmentation With Group Consistency Loss," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[25] D George, X Xie, and GKL Tam, "3D mesh segmentation via multi-branch 1D convolutional neural networks," *Graphical Models*, Vol. 96, pp. 1-10, 2018.

[26] J Ren, J Schneider, M Ovsjanikov, and P Wonka, "Joint Graph Layouts for Visualizing Collections of Segmented Meshes," *IEEE Transactions on Visualization and Computer Graphics*, Vol. 24, No. 9, pp. 2546 - 2558, 2018.

[27] A Poulenard, M Ovsjanikov, "Multi-directional geodesic neural networks via equivariant convolution," *ACM Transactions on Graphics (TOG)*, Vol. 37, No. 6, 2018.

[28] L Yi, H Su, X Guo, and LJ Guibas,"SyncSpecCNN: Synchronized Spectral CNN for 3D Shape Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[29] K Xu, VG Kim, Q Huang, and N Mitra, E Kalogerakis, "Data-driven shape analysis and processing," *SIGGRAPH ASIA*, pp. 1-38, 2016.

[30] E Kalogerakis, M Averkiou, S Maji, and S Chaudhuri,"3D Shape Segmentation With Projective Convolutional Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[31] P Skraba, M Ovsjanikov, F Chazal, and L Guibas, "Persistence-based segmentation of deformable shapes," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010.

[32] RM Rustamov, "Laplace-Beltrami eigenfunctions for deformation invariant shape representation," *Proceedings of the fifth Eurographics symposium on Geometry processing*, pp. 225-233, 2007.

[33] A Sharma, "Representation, Segmentation and Matching of 3D Visual Shapes using Graph Laplacian and Heat-Kernel," *Computer Vision and Pattern Recognition [cs.CV]*, Institut National Polytechnique de Grenoble - INPG, 2012.