The Visual Computer (2021) 37:1075–1088
https://doi.org/10.1007/s00371-020-01853-1

ORIGINAL ARTICLE

Automatic 3D virtual fitting system based on skeleton driving

Guangyuan Shi[1] · Chengying Gao[1] · Dong Wang[2] · Zhuo Su[1]

Published online: 25 May 2020

Abstract

To facilitate the input and improve the practicality of the existing virtual fitting systems, we propose a fully automatic 3D virtual fitting system to fit garment onto human models with various shapes and poses using a 2D full-body image and a 3D garment model. The proposed method constructs the 3D human model from the input 2D full-body image of the user by adopting the SMPLify method. To automatically position garment models onto human models with arbitrary postures, we present a 3D mesh segmentation method based on the discrete Reeb graph to accurately segment the different parts of a garment model, and a skeleton driving method based on mean curvature flow, which automatically adjusts the posture of the garment model according to the skeleton structural difference between the human model and the garment model. In addition, for the purpose of obtaining a more natural dress effect, we further adopt interpenetration removal and physical simulation for the deformed garment model. Compared to existing automatic 3D virtual fitting systems, the experimental results, we obtained based on the Leeds Sports Pose dataset, reveal that the proposed virtual fitting system is stable and effective.

Keywords Virtual fitting system · SMPLify · Mesh segmentation · Skeleton driving method

1 Introduction

The virtual fitting system is able to figure out the result of fitting a specific garment on a virtual human body in three-dimensional space, which can considerably improve the customer experience of online shopping and the efficiency of trying on dresses in physical stores. In the last decades, many 3D virtual fitting systems have been proposed. Some of them are not automatic and require human interaction [1,2]. Other methods [3–5], although automatic, require the input human model and garment models to have the same posture. To address these problems, some methods of fitting garment models onto human models with different postures have been proposed in recent years [6–11]. However, all these methods require a 3D human model as input, and some methods even require a 3D reference human body mesh model wearing a 3D garment. Because of the specific requirements of these methods for human models, it is difficult to popularize these virtual fitting methods. In addition, some garment fitting methods [10,12] are very time-consuming, because their energy equations constructed to solve core problems are relatively complex.

From our perspective, there are five necessary requirements for a qualified virtual fitting system. Firstly, the virtual fitting system is supposed to fit garment models onto human models with different postures. Secondly, the fitting results are required to reflect whether the input garment model fits the human model. Thirdly, the virtual fitting system is supposed to be automatic. Fourthly, the inputs of the automatic virtual fitting system should be simple, or to say, we can get the input information easily. Fifthly, the time for automatic fitting is supposed to be reasonable enough for real-time application.

Based on the five requirements above, we proposed a fully automatic virtual fitting system based on skeleton driving. The pipeline overview of the proposed automatic virtual fitting system is depicted in Fig. 1. Given a single full-body image, we first construct a SMPL human model [26]. Then, we segment the input 3D garment mesh model into primary body parts by our improved segmentation method. After that, we deform the garment model appropriately to fit the SMPL human model based on the structural difference of the human skeleton and garment skeleton. Finally, we dress

✉ Chengying Gao
mcsgcy@mail.sysu.edu.cn

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

[2] College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

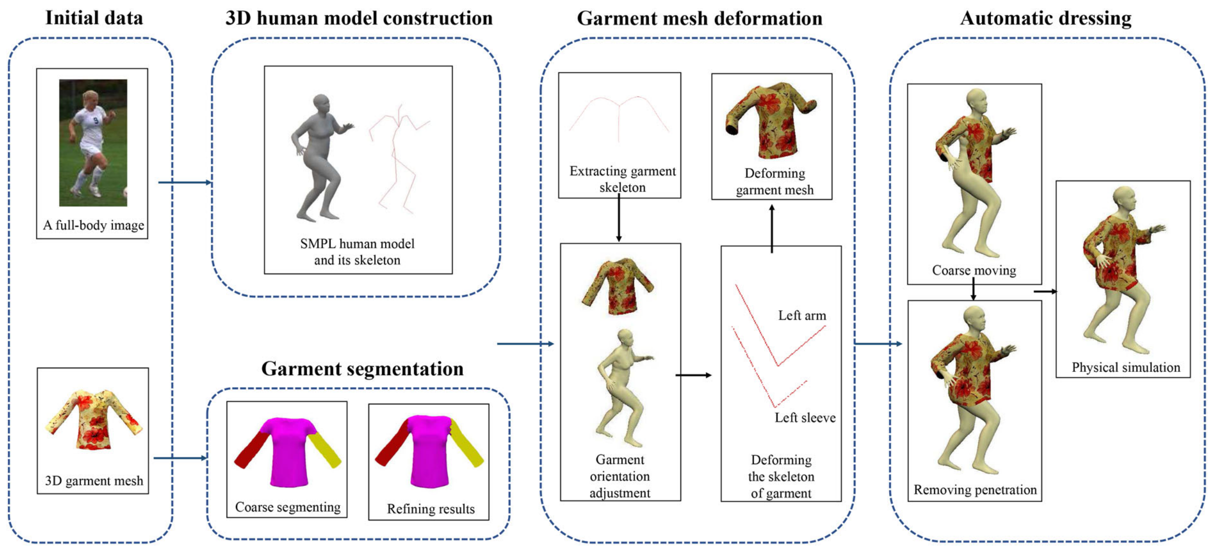


Fig. 1 Pipeline overview. Steps of the proposed automatic garment fitting system

the deformed garment model onto the SMPL human model automatically.

Our system has the following advantages: Firstly, the system does not require any user intervention. Secondly, the inputs of our system are just a single 2D full-body image and a 3D garment mesh; therefore, it is convenient for users to customize their body models to match their heights and shapes through our system. Thirdly, our system can be applied for human images with various postures and shapes. Fourthly, the automatic dressing component of our system is efficient. In conclusion, our system is practical, usable and user-friendly; therefore, it can be applied to many application scenarios such as online shopping and physical store. The contributions of our system are summarized as follows:

(1) We design a fully automatic 3D virtual fitting system. Different from existing 3D virtual fitting systems that need 3D human models as input, the system automatically fitting the given garment model onto the human model constructed from the input full-body image.

(2) We propose a 3D mesh segmentation method based on discrete Reeb graph. The proposed method improves the accuracy of segmentation in the upper arm and upper thigh of garment models and effectively avoids the distortion of fitting garment models onto the human model.

(3) We present a mean curvature flow-based method to fine-tune the posture of the input garment model according to the structural difference between the extracted skeletons of the garment model and the human model; therefore, we can easily fit the garment model onto the human model with any postures.

2 Related work

2.1 Constructing Human model from images

It is feasible to construct a human model according to human body parameters such as height and weight in a virtual try-on system [13,14]. However, compared to constructing a human model from images, these methods cannot specify the posture of the human model. It is a challenging problem to estimate the posture and shape of the human body from a single image. In order to solve this problem, many researchers use monocular image sequences or video sequences as reference [15–19], or explore alternative sensors [20,21]. Although these methods can obtain high accuracy, for a virtual fitting system, taking a single image as input is more practical.

Early works [22,23] use silhouettes or keypoints to estimate the parameters of the SCAPE model [24] from image, which often require some user intervention. Recently, Bogo et al. propose the first fully automatic approach, SMPLify [25], to predict 3D human shape based on SMPL model. They first extract 2D keypoints of the image by DeepCut. Then, they construct an objective function to fit the predicted 3D joints of SMPL model to the detected 2D keypoints using pose and shape priors to guide the optimization to obtain the key parameters of SMPL model. Beyond SMPLify, Lassner et al. [27] improve the fitting procedure using the estimated silhouettes. In general, these optimization-based methods require a good initialization, have slow running time and fail in some cases because of local minima. Therefore, recently, many regression-based methods have been proposed [28–35]. Taking the parts segmentation [32], keypoints and estimated silhouettes [29], raw pixels of the image [31] or hybrid annotations [35] as inputs, these methods regress SMPL parameters directly by applying different losses. These methods all achieve good performance in terms of accuracy and time consumption. However, these methods use various reprojection losses, which are weak supervisory signals. Therefore, instead of enforcing reprojection losses, Kolotouros et al. [34] optimize the regress shape with SMPLify [25] and use optimized parameters to supervise the network. Instead of regressing SMPL parameters directly, Kolotouros et al. [33] regress the locations of the mesh vertices with the use of a graph CNN.

2.2 Dressing virtual avatar

Existing methods for dressing virtual avatar can be mainly classified into three categories: garment transfer, garment fitting and learning to dress methods.

Garment transfer methods Given a target human model, garment transfer methods transfer the garment from the reference human model to the target human model, taking good advantage of the corresponding relationship between the reference and target human models. Li et al. [5] propose a method that transfers garment models based on the cross section of models and a constrained volumetric graph deformation. However, this method requires that the model is supposed to be in a rest pose. Brouet et al. [3] propose an automatic garment transfer method, which can be applied for human models with a different shape. The method regards garment transfer as an optimization problem and efficiently solves the problem. However, this method requires a reference human model which has the same posture with the target human model. In order to transfer the garment to the human model of any postures, many methods are proposed [6–9]. However, all of these methods require two human models and a garment mesh as inputs, which are not simple enough. Pons-Moll et al. [4] propose a novel method, which can estimate the garments and their motions from 4D scans of fully dressed people and retarget the garments to new human models. Although the inputs are simple and the results are natural, this method cannot transfer the garments to human models with different postures.

Garment fitting methods Compared with garment transfer methods, garment fitting methods do not require a reference human body model and directly fit the garment models onto the human models. Therefore, if there is not available reference human model, for example when the garment is obtained by a 3D laser scanner, garment fitting methods will be better. In the early stage, techniques of garment dressing mostly sew the 2D garment pieces or pattern on 3D human model [36–38]. Instead of sewing garments, a fitting method computes the transformation matrix of the garment model based on feature points [1]; however, this method is not fully automatic and cannot fit the garment model onto the human model with any postures. Jituo Li et al. [2] use the skeleton driving method to change the pose of the garment and then adjust the garment locally to get a more natural result. Although it can be applied for a human model with any postures, the procedure of adjusting the skeletons is not automatic. A fully automatic garment fitting method proposed by Tisserand et al. [10] minimizes the surface energy function to animate the human model to fit the garment model. Although this method can be applied for a human model with most postures, the consuming time of the minimization procedure is relatively long because the energy equation has many variables. Therefore, Nannan Wu et al. [11] improve this method based on the skeletons of human models. Although the improved method is more efficient and can be applied for various type of garments, similar to the work of Tisserand et al. [10], it cannot reflect whether the garment model fits the human model; in addition, according to the type of garment, the joints involved need to be selected manually.

Learning to dress methods Guan et al. [12] obtain a model learned from simulating the garment on the body with various shapes and postures. However, giving a different garment model, it is difficult to obtain the training set, and the tedious training process needs to be repeated, which is quite labor intensive. Clegg et al. proposed two methods [39,40] for animating human dressing. However, the system [39] requires the user to specify the sequence of actions in the dressing process for a dressing task, and the system [40] can only perform upper body tasks.

3 Human model construction

The 3D human model is an essential part of a 3D virtual fitting system. However, the 3D model of a user is not always available, so we integrate a 3D human modeling method based on the full-body image into our system. We construct our human body based on the SMPL model [26], a parameterized human model. The SMPL human model is defined by a function $M(\theta, \beta)$, where β is the shape parameter and θ is the pose parameter. The output of the function $M(\theta, \beta)$ is a watertight triangular mesh $\mathcal{M}$ with 6890 vertices and 13,766 faces. Therefore, with the use of the SMPL parameters θ and β, we can obtain the mesh of the target SMPL human model directly. Then, we can use the transformation matrix of each joint to calculate the positions of all transformed joints. By linking the joints orderly, we can obtain the skeleton structure of the target SMPL human model, which contains 23 joints and 22 edges.

We use SMPLify [25], the first fully automatic approach, to estimate SMPL model parameters from a single RGB image. This system first extracts the 2D joint locations of the image by DeepCut [42]. Then, for the purpose of minimizing the error between the detected 2D joints and the projected 3D joints of a SMPL human model, the system constructs an

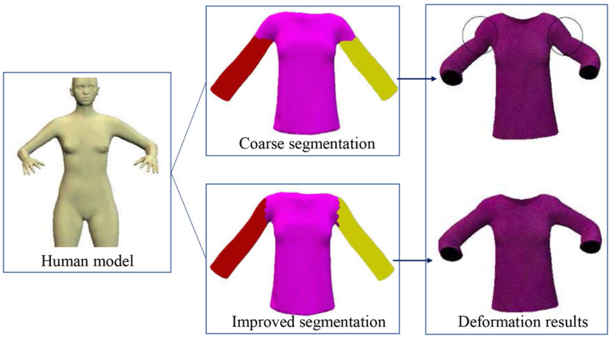


Fig. 2 Garment deformation results after segmenting garment with different segmentation methods

objective function. By minimizing the objective function, the system can obtain the key parameters of a SMPL model.

4 Garment segmentation

In the step of segmenting garment, we first segment the garment model using a discrete Reeb graph-based method [43], which is a method for segmenting human models. However, the segmentation results at the upper arms for clothes or upper thighs for trousers are not accurate enough; therefore, we refine the segmentation results after applying the discrete Reeb graph-based method.

4.1 Coarse segmentation

The discrete Reeb graph-based segmentation method [43], which is based on Morse functions, is applied for coarse segmentation. Because the input garment model is in a rest pose, we choose a Morse function, namely geodesic distance with a single source. The results of coarse segmentation are presented in Fig. 3a, b. It is obvious that the top of the upper thighs of the trouser and the top of the upper arms of the cloth are divided into the torso part. In fact, from the human physiological structure point of view, a more reasonable segmentation should refer to the location of the human joints. In many cases, this problem is not critical, but it will result in a distorted garment model mesh in the garment deformation step, as shown in Fig. 2.

4.2 Optimization

For the purpose of promoting the accuracy of segmentation, we adjust the segmentation boundaries to refine the segmentation results. In order to refine the segmentation results, we need to define new boundary surfaces, which are supposed to be on the joints of human model, as shown in Fig. 3b, d.

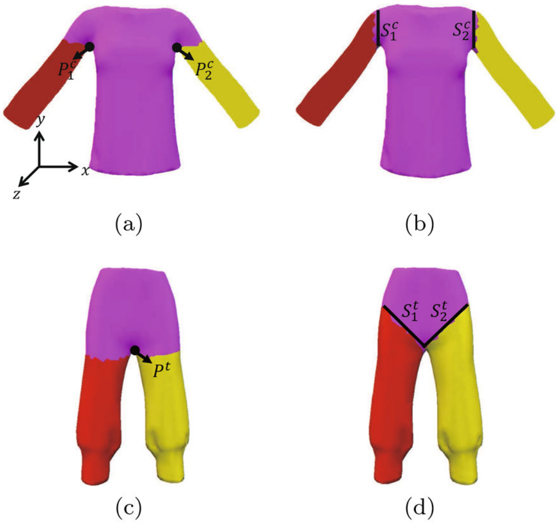


Fig. 3 Segmentation results of garment models. **a** Coarse Segmentation of a cloth model. **b** Refining the segmentation result of a cloth model. **c** Coarse segmentation of a trouser model. **d** Refining the segmentation result of a trouser model

Let $G_{\text{left\_sleeve}}$, $G_{\text{right\_sleeve}}$ and $G_{\text{c\_torso}}$ denote the vertices set of the left sleeve part, the right sleeve part and the torso part after coarse segmentation, respectively. In order to refine the segmentation results of a garment model, we define two boundary surfaces S_1^c and S_2^c between the arms and torso, as shown in Fig. 3b, and classify the vertices on the right side of the surface S_1^c and on the left side of the surface S_2^c as torso part. The keypoints of the garment model are denoted as P_1^c and P_2^c, as shown in Fig. 3a. According to the results of coarse segmentation, we can obtain the x coordinates of the keypoints P_1^c and P_2^c:

$$\begin{cases} P_1^c(x) = max(v(x)), & v \in G_{\text{right\_sleeve}} \\ P_2^c(x) = min(v(x)), & v \in G_{\text{left\_sleeve}}. \end{cases} \tag{1}$$

Assuming the garment model is facing forward, the surfaces S_1^c and S_2^c are both parallel to the Y–Z plane. Therefore, the surfaces S_1^c and S_2^c can be given as follows:

$$\begin{aligned} S_1^c: \quad x = P_1^c(x). \\ S_2^c: \quad x = P_2^c(x). \end{aligned} \tag{2}$$

Let $G_{\text{left\_leg}}$, $G_{\text{right\_leg}}$ and $G_{\text{t\_torso}}$ denote the vertices set of the left leg part, the right leg part and the torso part after coarse segmentation, respectively. For the purpose of adjusting the upper thighs of the trouser model, we define two boundary surfaces S_1^t and S_2^t between the thighs and torso, as shown in Fig. 3d, and divide the vertices below surfaces into the thigh parts. The keypoint of the trouser model is denoted as P_t, as shown in Fig. 3c. Then, the point P_t can be figured out based on the results of coarse segmentation as follows:

$$\begin{cases} P^t(x) = \frac{1}{N}\sum v(x), \quad v \in G_{\text{t\_torso}}, \\ P^t(y) = max(v(y)), \quad v \in G_{\text{right\_leg}} \bigcup G_{\text{left\_leg}}, \end{cases} \tag{3}$$

where N denotes the number of elements of set $G_{\text{t\_torso}}$. The slope k_l of the left straight line in Fig. 3d, which is mostly 1, is defined according to the type of trousers. The surfaces S_1^t and S_2^t can be figured out as follows:

$$\begin{aligned} S_1^t : \quad & -k_l x + y = -k_l P^t(x) + P^t(y), \\ S_2^t : \quad & k_l x + y = k_l P^t(x) + P^t(y). \end{aligned} \tag{4}$$

5 Garment deformation

In Sect. 4, we segment the input garment model into primary body parts. In this section, we deform each part of the garment model to obtain a deformed garment model with the same posture as the SMPL human model. Instead of extracting the skeletons of the human model and garment model in our previous work [41], we only need to extract the skeletons of all parts of the segmented garment using mean curvature flow (MCF)-based method [44]. The human skeleton can be obtained directly by connecting the joints of the constructed SMPL model. Then, we adjust the orientation of the garment based on the key joints of the garment and human models. After that, by comparing the difference in skeleton structure between the key parts of the garment and the corresponding parts of the human body, we obtained the deformed garment skeleton. Finally, we deform the garment mesh according to the deformed garment skeleton.

5.1 Skeleton extraction

The first step of our garment deformation method is skeleton extraction. The curve skeleton [45], which is a 1D skeletal shape contained edges and points in three-dimensional space, can represent the central axis of the 3D model. There are many methods for extracting curve skeleton from a 3D watertight model [44,46–48]. In our system, we use mean curvature flow-based method [44], which has a satisfying performance on cylindrical objects, to extract the skeletons of each part of the garment model. However, this method requires a watertight mesh as input, that is, to say, each segmented part of garment mesh needs to be preprocessed to obtain a watertight mesh, which should be approximated in shape to the original mesh. In our system, we adopt the hole-filling method [49] to fill the holes.

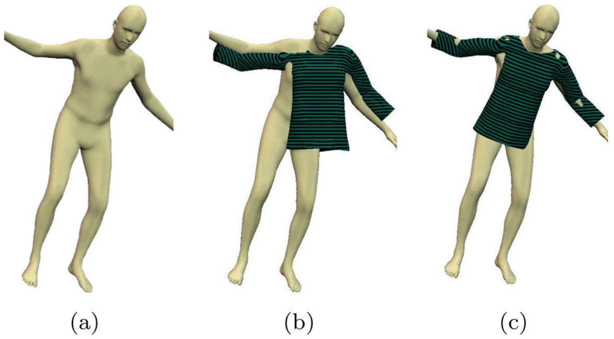


Fig. 4 **a** The input human model. **b** Deformation result of a garment without orientation adjustment. **c** Deformation result of a garment after orientation adjustment

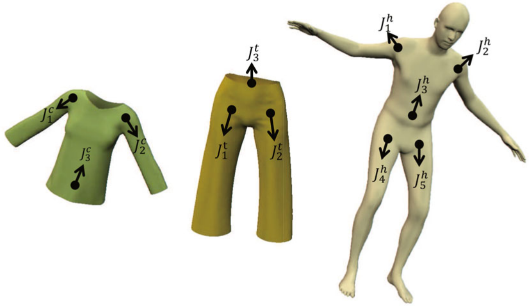


Fig. 5 Key joints of a cloth model, a trouser model and a human model (from left to right)

5.2 Orientation adjustment

In our previous work [41], we do not take into account the case where the torso orientation and the viewpoint are inconsistent, or to say, the human torso does not face us directly. To avoid the distortion problem in this case as shown in Fig. 4, we adjust the orientation of the garment model before deforming it.

The orientation of a garment model can be represented by a vertical vector and a horizontal vector. Therefore, according to the key joints of garment models and human model, as shown in Fig. 5, we can define the orientation of the cloth model, trouser model and human model as $D_c = \{\mathbf{v}_h^{\text{cloth}}, \mathbf{v}_v^{\text{cloth}}\}$, $D_t = \{\mathbf{v}_h^{\text{trouser}}, \mathbf{v}_v^{\text{trouser}}\}$ and $D_h = \{\mathbf{v}_h^{\text{hum}}, \mathbf{v}_v^{\text{hum}}\}$, respectively.

Taking the cloth model as an example, the orientation D_c can be calculated as follows:

$$\begin{cases} \mathbf{v}_h^{\text{cloth}} = norm(J_2^c - J_1^c), \\ \mathbf{v}_v^{\text{cloth}} = norm(J_3^c - \frac{J_2^c + J_1^c}{2}), \end{cases} \tag{5}$$

where J_1^c, J_2^c and J_3^c are the key joints of the cloth model, and function $norm(\cdot)$ represents the normalization of a vector. The orientations D_t and D_h are calculated in a similar way to D_c.

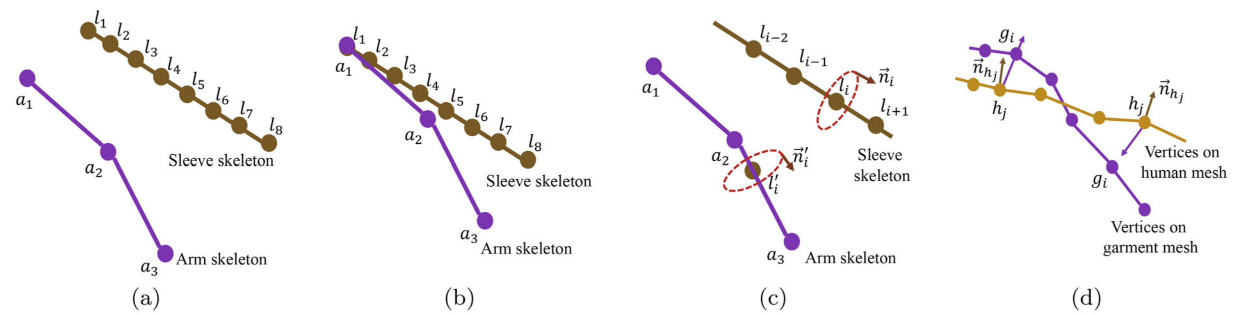


Fig. 6 Abstract representations. **a** A sleeve skeleton and an arm skeleton. **b** Making the first node of arm skeleton coincide with that of sleeve skeleton. **c** Deformation of garment vertices according to garment skeleton nodes. **d** Interpenetration phenomenon

There are two rotation matrices corresponding to the two vectors defined in the model orientation, which are denoted as $R_h \in \mathbb{R}^{3\times3}$ and $R_v \in \mathbb{R}^{3\times3}$. First of all, we apply the rotation matrix R_h to rotate the horizontal vector $\mathbf{v}_h^{\text{cloth}}$ to make it equal to $\mathbf{v}_h^{\text{hum}}$. The rotation axis $\mathbf{r_1}$ and rotation angle θ can be figured out as follows:

$$\mathbf{r_1} = \frac{\mathbf{v}_h^{\text{cloth}} \times \mathbf{v}_h^{\text{hum}}}{\|\mathbf{v}_h^{\text{cloth}} \times \mathbf{v}_h^{\text{hum}}\|}, \tag{6}$$

$$\theta = \arccos\left(\frac{\mathbf{v}_h^{\text{cloth}} \cdot \mathbf{v}_h^{\text{hum}}}{|\mathbf{v}_h^{\text{cloth}}| \times |\mathbf{v}_h^{\text{hum}}|}\right). \tag{7}$$

The matrix R_h can be figured out by the *Rodrigues' rotation formula*.

Then, we apply the rotation matrix R_v, which is applied for transferring the vector $R_v\mathbf{v}_v^{\text{cloth}}$ to $\mathbf{v}_v^{\text{hum}}$. R_v can be calculated in a similar way to R_h. Finally, we can obtain the final rotation matrix $R = R_v R_h$ for adjusting the garment orientation. The results of orientation adjustment are presented in Fig. 5.

5.3 Deformation of garment skeleton

The skeleton can reflect the posture of garment model to some extent. Therefore, in order to adjust the posture of garment model to fit the posture of human model, we first deform the garment skeleton according to the difference in skeleton structure between the garment model and human model.

As is shown in Fig. 6a, $L = \{l_1, \ldots, l_n\}$ represents the set of sleeve skeleton nodes, where n denotes the size of L, and $A = \{a_1, a_2, a_3\}$ represents the set of arm skeleton nodes. Let $f_{\text{seg}}(v_i, v_j)$ denote the distance between any two nodes of a skeleton, which is presented as follows:

$$f_{\text{seg}}(v_i, v_j) = \sum_{k=i}^{j-1} d(v_k, v_{k+1}), \quad 1 \le i < j, \tag{8}$$

where $d(v_i, v_j)$ denotes the Euclidean distance between nodes v_i and v_j in three-dimensional space.

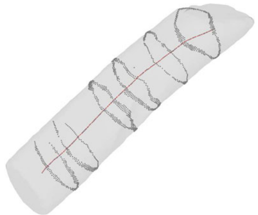


Fig. 7 Corresponding relationship between skeleton nodes and mesh

For deforming the sleeve skeleton, the first thing we need to do is moving the arm skeleton to make the first node of arm skeleton coincide with that of sleeve skeleton, as shown in Fig. 6b. For $\forall a \in A$,

$$a = a + (l_1 - a_1). \tag{9}$$

Then, we change the positions of $\{l_2, \ldots, l_n\}$ to make the sleeve skeleton and arm skeleton overlap. We can find two key nodes on the sleeve skeleton l_i and l_{i+1}, which satisfy

$$f_{\text{seg}}(l_1, l_i) \le f_{\text{seg}}(a_1, a_2) \le f_{\text{seg}}(l_1, l_{i+1}). \tag{10}$$

Now, the elbow joint a_2 is located on the skeleton segment $l_i l_{i+1}$. Let set $L' = \{l'_1, \ldots, l'_n\}$ denote the target position of nodes in L. The target nodes l'_2 to l'_i are located on the skeleton segment a_1a_2, and the target nodes l'_{i+1} to l'_n are located on the skeleton segment a_2a_3. Therefore, the elements of L' can be figured out as follows:

$$l'_k = \begin{cases} a_1 + \frac{f_{\text{seg}}(l_1, l_k)}{d(a_1, a_2)}(a_2 - a_1), & 1 < k \le i, \\ a_2 + \frac{f_{\text{seg}}(l_1, l_k) - d(a_1, a_2)}{d(a_2, a_3)}(a_3 - a_2), & i < k \le n. \end{cases} \tag{11}$$

5.4 Deformation of garment mesh

The skeleton extraction method [44] maintains the correspondence between the mesh vertices and the skeleton nodes in the extraction process, as shown in Fig. 7. In addition, this method guarantees the equation as follows:

$$\mathcal{M}(l_i) \bigcap \mathcal{M}(l_j) = \emptyset, \quad i \ne j \; and \; 1 \le i, j \le n, \tag{12}$$

where $\mathcal{M}(l_i)$ denotes the set of mesh vertices corresponding to l_i; that is to say, the mesh vertices corresponding to each node of extracted skeleton are disjoint.

We obtain the deformed skeleton of the sleeve L' in Sect. 5.3. The abstract representation of garment deformation is presented in Fig. 6c, where the red dotted lines represent the corresponding vertices of skeleton nodes l_i and l_i'. The garment mesh deformation refers to the change of $\mathcal{M}(l_i)$ to $\mathcal{M}(l_i')$ according the position change of l_i.

The vectors $\mathbf{n}_i$ and $\mathbf{n}_i'$, as shown in Fig. 6c, are the direction of the skeleton at nodes l_i and l_i', respectively, which can be calculated as follows:

$$\mathbf{n}_i = \begin{cases} l_{i+1} - l_i, & i = 1, \\ l_i - l_{i-1}, & i = n, \\ \frac{l_{i+1} - l_{i-1}}{2}, & 1 < i < n, \end{cases} \tag{13}$$

$$\mathbf{n}_i' = \begin{cases} a_2 - a_1, & l_i \text{ is located on } a_1a_2, \\ a_3 - a_2, & l_i \text{ is located on } a_2a_3. \end{cases} \tag{14}$$

According to the *Rodrigues' rotation formula*, we can obtain the rotation matrix $Q_i \in \mathbb{R}^{3\times3}$ for rotating $\mathbf{n}_i$ to $\mathbf{n}_i'$. Then, we expand the rotation matrix Q_i to $Q_i' \in \mathbb{R}^{4\times4}$, which is given as follows:

$$Q_i' = \begin{bmatrix} Q_i & 0 \\ 0 & 1 \end{bmatrix}. \tag{15}$$

We define two translation matrices $T_1 \in \mathbb{R}^{4\times4}$ and $T_2 \in \mathbb{R}^{4\times4}$ for translating the node l_i to the coordinate origin and translating the node from the coordinate origin to l_{i+1}'. The transformation matrix for the vertices in $\mathcal{M}(l_i)$ is $T_2 Q_i' T_1$. Finally, we use Laplacian surface deformation method [51] to ensure continuity between adjacent segments.

6 Automatic dressing

Given a SMPL human model and a deformed garment mesh, we first move the deformed garment mesh to the corresponding part of the human model coarsely based on the feature joints. Then, we remove the interpenetration between the garment model and human model. Finally, for the purpose of generating natural shape, which means that the simulation results basically need to comply with the physical laws, we carry out physical simulation for the garment model based on physical force.

6.1 Coarse move

According to the skeleton of SMPL human model constructed by SMPLify [25] and the skeletons of garment models extracted in Sect. 5.1, we can obtain the key joints of garment models and human model, as shown in Fig. 5. Then, the translation vectors v_c for a cloth model and v_t for a trouser model can be figured out as follows:

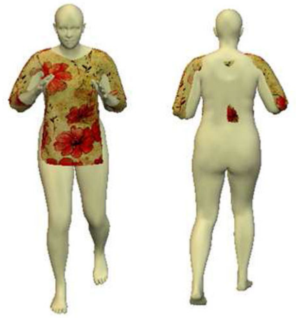


Fig. 8 Interpenetration between a cloth model and a human model

$$v_c = \frac{J_1^h + J_2^h}{2} - \frac{J_1^c + J_2^c}{2}, \tag{16}$$

$$v_t = \frac{J_4^h + J_5^h}{2} - \frac{J_1^t + J_2^t}{2}. \tag{17}$$

6.2 Removal of interpenetration

Because the garment models are elastic, in the natural state, the shapes of garment models are inconsistent with the shape of the human model. Therefore, we need to remove the interpenetration between the human model and garment model, as shown in Fig. 8, after the coarse move step. The method we adopt for removing interpenetration is similar to the method in [12]. The abstract representation of the interpenetration phenomenon is presented in Fig. 6d. For any vertex g_i on the garment mesh, we denote the vertex closest to it on the human mesh as h_j. We define a function $f_{\text{dot}}(i, j)$, which represents the inner product of vector $\mathbf{n}_{h_j}$ and $(g_i - h_j)$:

$$f_{\text{dot}}(i, j) = (g_i - h_j) \cdot \mathbf{n}_{h_j}, \tag{18}$$

where $\mathbf{n}_{h_j}$ denotes the normal for vertex h_j. It is obvious that if $f_{\text{dot}}(i, j) < 0$, interpenetration happens; otherwise, not. Therefore, we can define an objective function

$$E(G, H) = \lambda_1 \nu(G, H) + \lambda_2 \omega(H), \tag{19}$$

where λ_1, λ_2 are weights; G is the garment mesh; H is the human mesh.

To ensure that the vertices on garment mesh are outside the human mesh, we define a term $\nu(G, H)$ to penalize interpenetration phenomenon:

$$\nu(G, H) = \sum_{i \in I \wedge (i,j) \in \mathcal{T}} (\kappa - f_{\text{dot}}(i, j)), \tag{20}$$

where κ denotes the distance between the garment mesh and human mesh; I is the set of garment vertices which penetrate

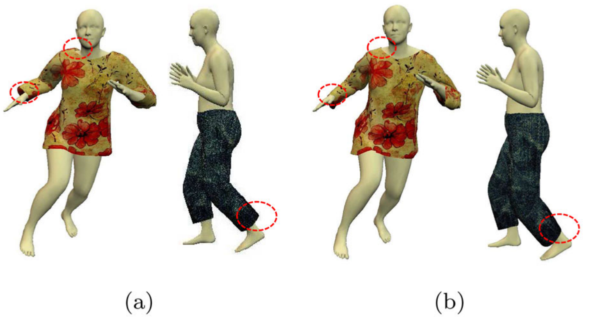


Fig. 9 **a** Unnatural dressing results without applying physical simulation. **b** Fitting results after applying physical simulation

the human mesh; $\mathcal{T}$ denotes the correspondence between each vertex g_i on garment mesh and its closest vertex h_j on human mesh.

The term $\omega(H)$, which is defined for maintaining the smoothness of the surface of the garment mesh, makes the deformation of a vertex similar to the deformation of its adjacent vertex. This term is presented as follows:

$$\omega(H) = \sum_{i \in H} \left\| (g_i' - g_i) - \frac{1}{\sum\limits_{k \in N_i} \frac{1}{d(g_k, g_i)}} \sum_{j \in N_i} \frac{(g_j' - g_j)}{d(g_j, g_i)} \right\|^2, \tag{21}$$

where N_i denotes the set of vertices adjacent to the vertex g_i; g' denotes the vertex before removing interpenetration; $d(g_i, g_j)$ denotes the Euclidean distance between vertex g_i and g_j.

Finally, we adopt a linear least squares solver to minimize the objective function $E(G, H)$.

6.3 Physical simulation

The deformation of the garment model is rigid and based on geometry. For the reason that the physical characteristics of the garment are not considered, the dressing results are unnatural, as shown in Fig. 9a. The parts of the garment model indicated by red dotted circle are floating, which are unnatural. In addition, the garment models in the figure seem stiff. The fitting results after applying physical simulation are presented in Fig. 9b.

For the purpose of obtaining more natural shape, we simulate the garment models by mass–spring system. The edges of the mesh of garment model represent the tension springs, and the rest length of the tension spring is the initial edge length. The resultant force on particle g_i is

$$f_i = f_i^g + f_i^d + f_i^b + f_i^{db} + f_i^s + f_i^{ds}, \tag{22}$$

where f_i^g denotes gravity; f_i^d denotes global damping force; f_i^b denotes bending force; f_i^{db} denotes the damping force of bending spring; f_i^s denotes elastic force; f_i^{ds} denotes the damping force of elastic spring.

Let $f_i(t)$ denote the resultant force on particle g_i at time t. The velocity and position of particle g_i at time t can be calculated as follows:

$$v_i(t) = v_i(0) + \int_{t_0}^{t} \frac{f_i(t)}{m_i} dt \tag{23}$$

$$s_i(t) = s_i(0) + \int_{t_0}^{t} v_i(t) dt \tag{24}$$

Finally, with the use of the Runge–Kutta fourth-order integral method, the positions of the vertices during the numerical simulation can be calculated efficiently and stably.

7 Experimental results

For the purpose of evaluating the proposed virtual fitting system, we take several full-body images from Leeds Sports Pose (LSP) dataset [50] and several types of garment from 3D modeling software as inputs. The images we used are denoted as $I = \{I_1, \ldots, I_8\}$, as shown in Fig. 10. Our proposed virtual fitting system can dress different types of garment on human model constructed from the input image automatically. In most cases, we obtain satisfying fitting results; however, in some cases, we cannot get good results. We analyze these unsuccessful cases and the limitations of our system in the **Limitation** section.

7.1 Results analysis

Same garment with different human postures The fitting results of dressing the same cloth on human models constructed from different images are presented in Fig. 10. Regardless of the posture in the input image, the system can get an ideal natural dressing result. It is obvious that the garment mesh and human mesh do not penetrate each other; in addition, after physical simulation, the garment is no longer stiff and not floating. What's more, the system can dress the same garment on human models with different shapes, which reflects the elasticity of the garment.

Same human image with different garments We manage to dress various types of garments on the constructed human model, as shown in Fig. 11. The fitting results are natural; therefore, our proposed virtual fitting system is suitable for various types of garments.

Different perspectives For the purpose of confirming whether our fitting results are satisfying, we present different perspectives of the fitting result of fitting a garment model on a human model in Fig. 12. Our proposed system can present

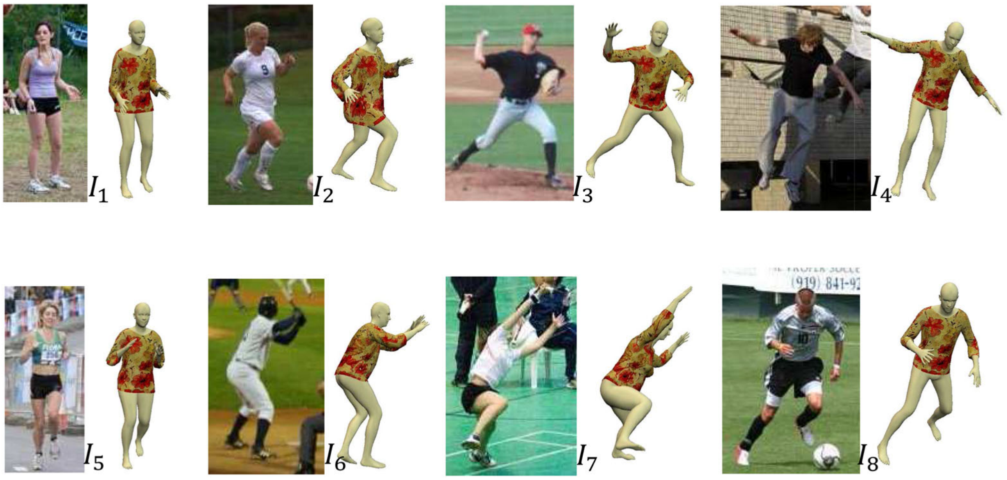


Fig. 10 Fitting a 3D garment model onto the human bodies constructed from several images of Leeds Sports Dataset

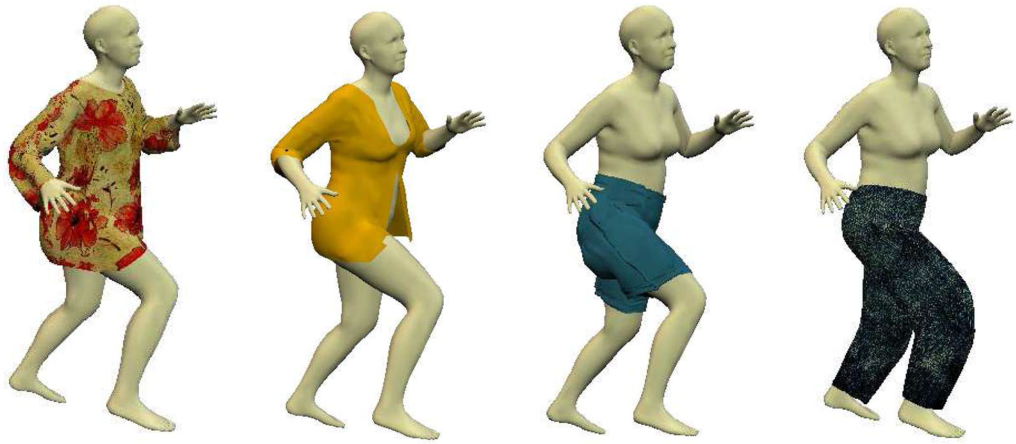


Fig. 11 The fitting results of dressing four types of garments on the human model constructed from image I_2

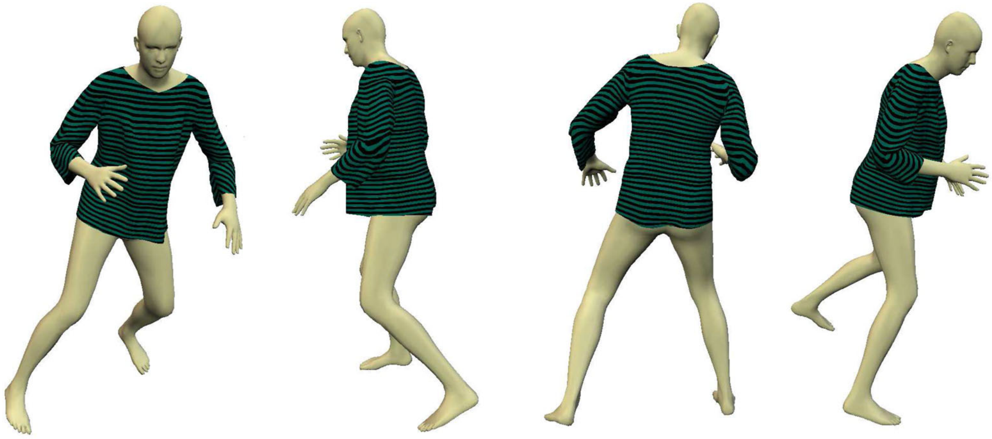


Fig. 12 Different perspectives of the fitting results of dressing a cloth on the human model constructed from image I_8

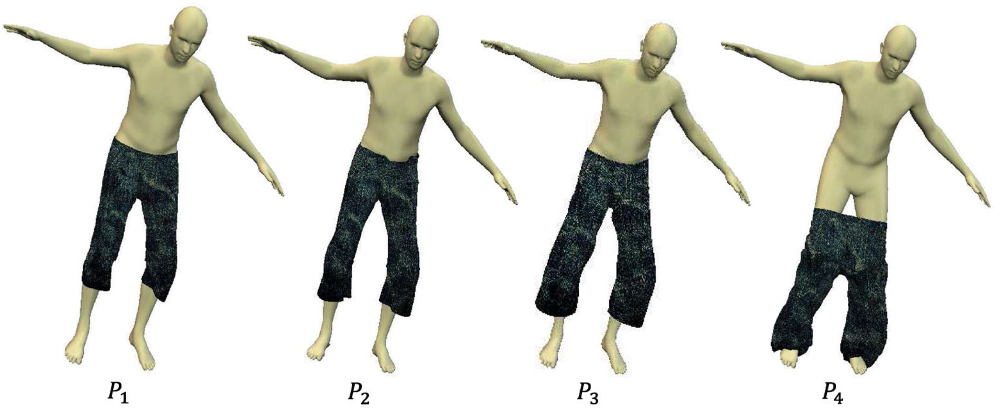


Fig. 13 The fitting results of dressing four different sizes of trousers on the human model constructed from image I_4

the fitting results in 3D space, which is convenient for users to evaluate the fitting result from various perspectives.

Same human image with different sizes of garment In Fig. 13, trousers of four different sizes are dressed on the same human model constructed from the image I_4. We denote the sizes of these four trousers as P_1, P_2, P_3 and P_4, which are incremental, i.e., $P_1 < P_2 < P_3 < P_4$. According to the visual appearance, it seems that the size P_2 trouser is the best fit. The size P_1 trouser seems like a trifle small and tight. The legs of size P_3 trouser are a trifle loose. The size P_4 trouser is so loose that it cannot be worn on the human model; therefore, the trousers slide down during the physical simulation.

Different resolution clothes In Fig. 14, we present the fitting results of dressing clothes with different resolutions. According to the visual effects, the result of the T-shirt with higher resolution is more natural; however, the fitting result of the lower-resolution T-shirt is also satisfactory. Therefore, for high-resolution clothes, we can simplify the mesh properly to reduce the time spent on fitting.

Custom full-body image In Fig. 15, we present the fitting result of dressing garments on the human model constructed from a user-input custom full-body image. The garments have similar styles and colors to the garments in the input image. Although the lights are different in the two scenes, it turns out that the visual effect of the dressing result is relatively natural and similar to the image.

7.2 Performance evaluation

All experiments are conducted on a computer equipped with an Intel 4 × 2.50 GHz CPU, and the calculation time of the main steps of garment fitting is given in Table 1. According to this table, the interpenetration removal and garment deformation contribute to most of the computational cost.

The consuming time of garment segmentation step is mainly determined by the number of the vertices and topology of the garment model; therefore, shirt 2 having more vertices than shirt 1 needs to spend more time to segment, and pants having more vertices than shirts spend less time to segment because of their simpler topology.

The time it takes to deform the garment is independent of the posture of the human model; therefore, the time spent on deforming shirt 2 to fit the human model constructed from image I_2 and I_4 is substantially equal. In fact, the time it takes to deform the garment is mainly determined by the number of vertices. It is obvious that it takes more time to deform the garment with more vertices. In addition, the topology of the garment also determines the time for the garment deformation step, because the topology of the garment model is the main factor affecting the time for skeleton extraction.

The time for removing interpenetration is determined by the number of elements of the set of garment vertices which penetrate the human mesh; therefore, for the same human model constructed from image I_4, it takes less time for pant 1 with larger size to remove interpenetration, because a larger pant means fewer vertices that penetrate human mesh.

7.3 Comparisons analysis

We compare our proposed system with related works in recent years from five aspects: system input, speed, applicable human postures, applicable garment types, whether fully automatic, as shown in Table 2. It is obvious that the input of our proposed system is the easiest to get, which means our system is more practical. What's more, our system is fully automatic compared to other garment fitting methods [2,10,11]. Although the system [4] can obtain a dressing result that is substantially the same as the dressing result in real life, it cannot be applied to a human model with any postures.

The systems [2,4,9–11] generally contain many modules, but they always lack specific implementation details in some steps. It is difficult to reproduce their works, and we cannot obtain the overall computation complexities of these sys-

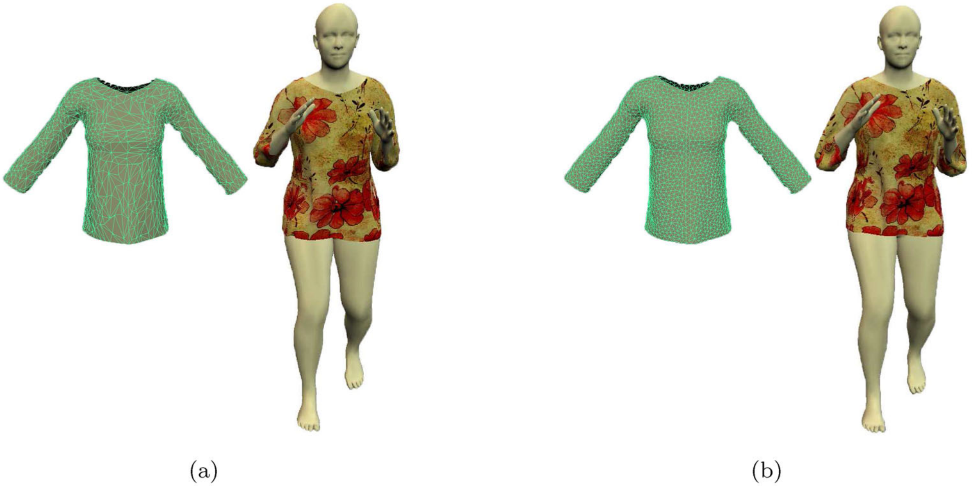

(a) (b)

Fig. 14 The fitting results of dressing clothes with different resolutions. **a** The fitting result of a T-shirt with 1751 vertices and 5247 edges. **b** The fitting result of a T-shirt with 3550 vertices and 10,644 edges

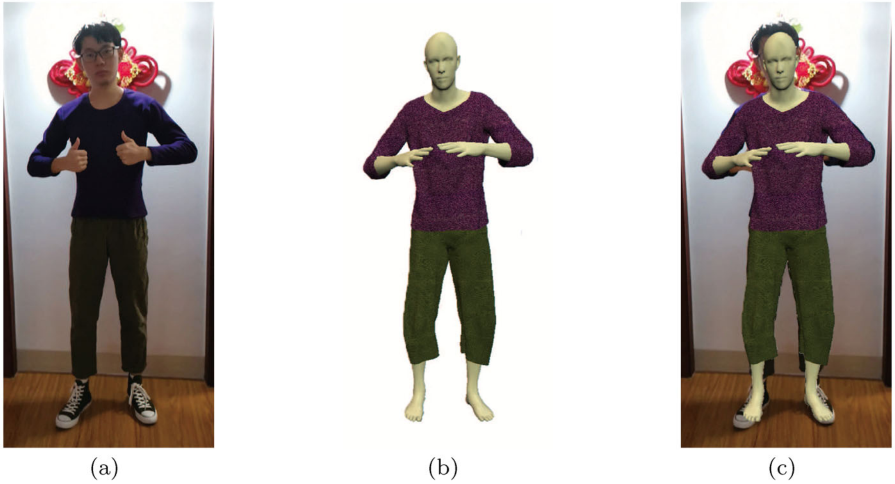

(a) (b) (c)

Fig. 15 The fitting results of dressing garments on the human model constructed from a custom full-body image. **a** The input image. **b** The dressing result. **c** The overlapped image

Table 1 Execution time of garment fitting

Garment	Image	Garment vertices/faces	Garment segmentation (ms)	Garment deformation (ms)	Penetration removal (ms)	Total (ms)
Dress	I_1	5875/11593	788	3552	5071	9422
Shirt 1	I_2	3106/9312	223	1568	3984	5781
Shirt 2	I_2	3550/7096	355	1957	3754	6070
Shirt 2	I_4	3550/7096	349	1924	4075	6355
Pant 1 (P_1)	I_4	4200/8396	271	3619	7011	10904
Pant 1 (P_2)	I_4	4200/8396	266	3630	4534	8439
Pant 1 (P_3)	I_4	4200/8396	279	3637	2929	6861
Pant 2	I_5	3062/6120	159	2859	3575	5599
Shorts 1	I_5	8196/16388	987	4584	7821	13399

tems. Therefore, we can only compare the critical processes of our system with those of recently proposed similar systems [10,11]. Similar to our system, these systems [10,11] mainly consist of garment fitting, interpenetration removal and physical simulation. Garment fitting is one of the most critical steps. Therefore, we conduct complexity analysis and compare ours with the state of the art [10,11] briefly.

The computational complexity of the optimization step for the method [10] is $O(K \cdot n_H n_G)$, where n_H is the number of the vertices of the human mesh; n_G is the number of the vertices of the garment mesh; K is the number of iterations of the optimization step. Nannan Wu et al. [11] propose a faster method. Its computational complexity of the optimization step is $O(K \cdot n_H)$. By contrast, the garment fitting

Table 2 Comparison of similar works

Reference	Input	Applicable human postures	Applicable garment types	Whether fully automatic
[9]	Reference human model wearing a garment mesh and a target human model	Any postures	Most garment types	Fully automatic
[4]	The 4D scans of a reference human model wearing a garment mesh and a target human model	Postures sequences consistent with the reference human pose sequences	Most garment types	Fully automatic
[11]	A target human model and a garment mesh	Any postures	Most garment types	The joints used in the optimization procedure require manual selection
[10]	A target human model and a garment mesh	Any postures	Most garment types	The joints used in the optimization procedure require manual selection
[2]	A target human model and a garment mesh	Any postures	Most garment types	The adjustment of skeleton is not automatic
Ours	A full-body image and a garment mesh	Any postures	Most garment types	Fully automatic

step of our method does not require an optimization step and the computational complexity is $O(n_H logn_H)$. Note that the computation time of our method is independent of the resolution of the human model; therefore, our skeleton-driven garment fitting method can be applied for a high-resolution human model with tens of thousands of vertices.

7.4 Limitations

There are mainly four limitations of our proposed system. Firstly, if the torso of the human model is greatly bent, the torso of the garment model will be unnatural after the garment model is fitted. The reason lays in the fact that the bending of the garment torso is very different from that of the human torso after garment deformation, which will distort the garment model after the removal of interpenetration. However, in general, the torso of the human body is straight or slightly bent. One possible solution is to extract the skeleton of the torso part of the garment model and deform this part based on the skeleton structural difference between the torso part of the garment model and human model.

In addition, if the cross section of the torso of the garment model is significantly different from the cross section of the torso of the human model in shape and size, the garment will be distorted after the removal of interpenetration because of the big difference between the human model and garment model. From our perspective, a possible avenue for addressing this problem is to choose the appropriate size garment by comparing the size and shape differences between the cross section of the garment model and the human model.

What's more, the system SMPLify [25] we applied for constructing the human model from a single full-body image is not efficient and accurate enough, especially in shape. SMPLify is an optimization-based method to fit SMPL to 2D keypoint detections. Similar to other optimization-based methods, its running time is low and it is not accurate enough if the initialization is unsatisfying. However, some more precise and efficient regression-based methods based on SMPL model have been proposed such as [33–35]. The body size parameters error of the human model constructed by these methods has been reduced to 31 mm [35]. Since our system is easy to be extended to any SMPL-based methods to construct the human model from an image, we will improve the human modeling module with these proposed systems.

Finally, the fitting results are relatively natural, which means that the results generally meet the basic physical rules; however, we cannot prove that our fitting results are satisfactory to the potential consumers, which determines the application value of our method. Therefore, in future research, we need to investigate the satisfaction of potential users by collecting their evaluations of their dressing results, which is similar to the result in Fig. 15c.

8 Conclusions

In this paper, we present an automatic virtual fitting system for fitting the input garment model onto the human model constructed from the input full-body image. Given a full-body image, the system first constructs a SMPL human model from the image. Then, the system segments the garment model and changes the posture of the garment model to fit the posture of the human model. Finally, the system put the deformed garment on the human model; in addition, to obtain a more natural and authentic fitting result, the system removes the interpenetration and applies physical simulation for the garment. According to the fitting results, our system is feasible, stable and efficient.

Acknowledgements This work was supported by the Natural Science Foundation of Guangdong Province, China (Grant No. 2019A1515011 075), National Natural Science Foundation of China (Grant No. 61972433, 61872394) and Fundamental Research Funds for the Central Universities (19lgjc11).

References

1. Li, Z., Jin, X., Barsky, B., Liu, J.: 3D clothing fitting based on the geometric feature matching. In: 2009 11th IEEE International Conference on Computer-Aided Design and Computer Graphics, pp. 74-80. IEEE (2009)
2. Li, J., Ye, J., Wang, Y., Bai, L., Lu, G.: Fitting 3D garment models onto individual human models. Comput. Graph. **34**(6), 742–755 (2010)
3. Brouet, R., Sheffer, A., Boissieux, L., Cani, M. P.: Design preserving garment transfer. ACM Trans. Graph. **31**(4) (2012)
4. Pons-Moll, G., Pujades, S., Hu, S., Black, M.J.: ClothCap: seamless 4D clothing capture and retargeting. ACM Trans. Graph. (TOG) **36**(4), 73 (2017)
5. Li, J., Lu, G.: Customizing 3D garments based on volumetric deformation. Comput. Ind. **62**(7), 693–707 (2011)
6. Lee, Y., Ma, J., Choi, S.: Automatic pose-independent 3D garment fitting. Comput. Graph. **37**(7), 911–922 (2013)
7. Narita, F., Saito, S., Kato, T., Fukusato, T., Morishima, S.: Pose-independent garment transfer. In: SIGGRAPH Asia 2014 Posters, p. 12 (2014)
8. Narita, F., Saito, S., Kato, T., Fukusato, T., Morishima, S.: Texture preserving garment transfer. In: ACM SIGGRAPH 2015 Posters, p 91 (2015)
9. Jiang, L., Ye, J., Sun, L., Li, J.: Transferring and fitting fixed-sized garments onto bodies of various dimensions and postures. Comput.-Aided Des. **106**, 30–42 (2019)
10. Tisserand, Y., Cuel, L., Magnenat-Thalmann, N.: Automatic 3D garment positioning based on surface metric. Comput. Anim. Virtual Worlds **28**, e1770 (2017)
11. Wu, N., Deng, Z., Huang, Y., Liu, C., Zhang, D., Jin, X.: A fast garment fitting algorithm using skeleton-based error metric. Comput. Anim. Virtual Worlds **29**(3–4), e1811 (2018)
12. Guan, P., Reiss, L., Hirshberg, D.A., Weiss, A., Black, M.J.: DRAPE: dressing any person. ACM Trans. Graph. (TOG) **31**(4), 1–10 (2012)
13. Wang, C.C.: Parameterization and parametric design of mannequins. Comput.-Aided Des. **37**(1), 83–98 (2005)
14. Baek, S.Y., Lee, K.: Parametric human body shape modeling framework for human-centered product design. Comput.-Aided Des. **44**(1), 56–67 (2012)
15. Xu, W., Chatterjee, A., Zollhöfer, M., Rhodin, H., Mehta, D., Seidel, H.P., Theobalt, C.: Monoperfcap: Human performance capture from monocular video. ACM Trans. Graph. (TOG) **37**(2), 27 (2018)
16. Alldieck, T., Kassubeck, M., Wandt, B., Rosenhahn, B., Magnor, M.: Optical flow-based 3d human motion estimation from monocular video. In: German Conference on Pattern Recognition, pp. 347–360. Springer (2017)
17. Grest, D., Herzog, D., Koch, R.: Human model fitting from monocular posture images. In: Proceedings of the VMV, pp. 665–1344 (2005)
18. Balan, A. O., Sigal, L., Black, M. J., Davis, J. E., Haussecker, H. W.: Detailed human shape and pose from images. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8. IEEE (2007)
19. Jain, A., Thormählen, T., Seidel, H.P., Theobalt, C.: Moviereshape: tracking and reshaping of humans in videos. ACM Trans. Graph. (TOG) **29**(6), 148 (2010)
20. von Marcard, T., Rosenhahn, B., Black, M.J., Pons-Moll, G.: Sparse inertial poser: automatic 3d human pose estimation from sparse imus. Comput. Graph. Forum. **36**, 349–360 (2017)
21. Weiss, A., Hirshberg, D., Black, M. J.: Home 3D body scans from noisy image and range data. In: 2011 International Conference on Computer Vision, pp. 1951–1958. IEEE (2011)
22. Sigal, L., Balan, A., Black, M. J.: Combined discriminative and generative articulated pose and non-rigid shape estimation. In: Advances in Neural Information Processing Systems, pp. 1337–1344 (2008)
23. Guan, P., Weiss, A., Balan, A. O., Black, M. J.: Estimating human shape and pose from a single image. In: 2009 International Conference on Computer Vision, pp. 1381–1388. IEEE (2009)
24. Anguelov, D., Srinivasan, P., Koller, D., Thrun, S., Rodgers, J., Davis, J.: SCAPE: shape completion and animation of people. ACM Trans. Graph. (TOG) **24**(3), 408–416 (2005)
25. Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., Black, M. J.: Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In: European Conference on Computer Vision, pp. 561–578 (2016)
26. Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., Black, M.J.: SMPL: a skinned multi-person linear model. ACM Trans. Graph. (TOG) **34**(6), 248 (2015)
27. Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M. J., Gehler, P. V.: Unite the people: closing the loop between 3d and 2d human representations. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 6050–6059 (2017)
28. Tan, V., Budvytis, I., Cipolla, R.: Indirect deep structured learning for 3d human body shape and pose prediction. BMVC **3**, 6 (2017)
29. Pavlakos, G., Zhu, L., Zhou, X., Daniilidis, K.: Learning to estimate 3D human pose and shape from a single color image. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 459–468 (2018)
30. Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., Schmid, C.: Bodynet: Volumetric inference of 3d human body shapes. In: European Conference on Computer Vision, pp. 20–36 (2018)
31. Kanazawa, A., Black, M. J., Jacobs, D. W., Malik, J.: End-to-end recovery of human shape and pose. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 7122–7131 (2018)
32. Omran, M., Lassner, C., Pons-Moll, G., Gehler, P., Schiele, B.: Neural body fitting: unifying deep learning and model based human pose and shape estimation. In: International Conference on 3D Vision (3DV), pp. 484–494 (2018)
33. Kolotouros, N., Pavlakos, G., Daniilidis, K.: Convolutional Mesh regression for single-image human shape reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 4501–4510 (2019)
34. Kolotouros, N., Pavlakos, G., Black, M. J., Daniilidis, K.: Learning to Reconstruct 3D Human Pose and shape via model-fitting in the loop. In: IEEE International Conference on Computer Vision, pp. 2252–2261 (2019)
35. Rong, Y., Liu, Z., Li, C., Cao, K., Loy, C. C.: Delving deep into hybrid annotations for 3D Human recovery in the wild. In: IEEE International Conference on Computer Vision, pp. 5340–5348 (2019)
36. Fuhrmann, A., Groß, C., Luckas, V., Weber, A.: Interaction-free dressing of virtual humans. Comput. Graph. **27**(1), 71–82 (2003)
37. Volino, P., Cordier, F., Magnenat-Thalmann, N.: From early virtual garment simulation to interactive fashion design. Comput.-Aided Des. **37**(6), 593–608 (2005)

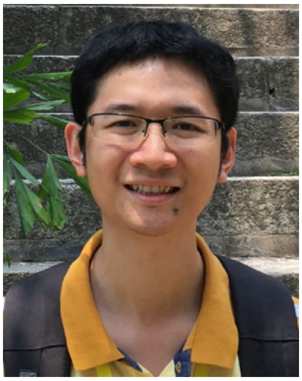
38. Metaaphanon, N., Kanongchaiyos, P.: Real-time cloth simulation for garment CAD. In Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, pp. 83–89 (2005)
39. Clegg, A., Tan, J., Turk, G., Liu, C.K.: Animating human dressing. ACM Trans. Graph. (TOG) **34**(4), 116 (2015)
40. Clegg, A., Yu, W., Tan, J., Liu, C. K., Turk, G.: Learning to dress: synthesizing human dressing motion via deep reinforcement learning. In: SIGGRAPH Asia 2018, pp. 179 (2018)
41. Cai, H., Shi, G., Gao, C., Wang, D.: Automatic 3D garment fitting based on skeleton driving. In: Pacific rim conference on multimedia, pp. 267–277 (2018)
42. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P. V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: IEEE conference on computer vision and pattern recognition, pp. 4929–4937 (2016)
43. Werghi, N., Xiao, Y., Siebert, J.P.: A functional-based segmentation of human body scans in arbitrary postures. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **36**(1), 153–165 (2006)
44. Tagliasacchi, A., Alhashim, I., Olson, M., Zhang, H.: Mean curvature skeletons. Comput. Graph. Forum **31**(5), 1735–1744 (2012)
45. Cornea, N.D., Silver, D., Min, P.: Curve-skeleton properties, applications, and algorithms. IEEE Trans. Visual. Comput. Graph. **13**, 530–548 (2007)
46. Au, O.K.C., Tai, C.L., Chu, H.K., Cohen-Or, D., Lee, T.Y.: Skeleton extraction by mesh contraction. ACM Trans. Graph. (TOG) **27**(3), 44 (2008)
47. Wang, Y.S., Lee, T.Y.: Curve-skeleton extraction using iterative least squares optimization. IEEE Trans. Visual. Comput. Graph. **14**(4), 926–936 (2008)
48. Dey, T.K., Sun, J.: Defining and computing curve-skeletons with medial geodesic function. Symp. Geom. Process. **6**, 143–152 (2006)
49. Eberly, D.: Triangulation by ear clipping. Geom. Tools. **2002–2005** (2008)
50. Johnson, S., Everingham, M.: Clustered Pose and nonlinear appearance models for Human Pose estimation. BMVC **2**, 5 (2010)
51. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H. P.: Laplacian surface editing. In: 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, pp. 175–184 (2004)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Guangyuan Shi received the B.S. degree from School of Data & Computer Science, Sun Yat-sen University in 2019. His research interests include computer graphics and image processing.

Chengying Gao is an associate professor of School of Data and Computer Science, Sun Yat-sen University. She received the Ph.D. degree in computer science from School of Information Science and Technology, Sun Yat-sen University, in 2003. Her research interests include computer graphics and visual media computing.

Dong Wang is an associate professor of College of Mathematics and Informatics, South China Agricultural University. She received the Ph.D. degree in computer science from City University of Hong Kong in 2012. Her research interests include computer graphics and image processing.



Zhuo Su is an associate professor of School of Data and Computer Science, Sun Yat-sen University. He received the Ph.D. degree in computer science from School of Information Science and Technology, Sun Yat-sen University, in 2014. His research interests include computer graphics and visual media computing.