Customizing 3D garments based on volumetric deformation

Jituo Li a,b,c,*, Guodong Lu a

a Institute of Engineering and Computer Graphics, Department of Mechanical Engineering, Zhejiang University, Hangzhou 310027, PR China
b Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai, PR China
c Digital Interactive Media Lab, Institute of Automation, Chinese Academy of Sciences, PR China

A R T I C L E   I N F O

Article history:
Received 12 January 2010
Received in revised form 29 March 2011
Accepted 7 April 2011
Available online 6 May 2011

Keywords:
3D garment customization
Volumetric deformation
Cross-sections
Tetrahedral mesh

A B S T R A C T

Improving the reusability of design results is very important for garment design industry, since designing an elegant garment is usually labor-intensive and time-consuming. In this paper, we present a new approach for customizing 3D garment models. Our approach can transfer garment models initially designed on a reference human model onto a target human model. To achieve this goal, firstly a spatial mapping between the two human models is established with the shape constraints of cross-sections. Secondly, the space around the clothed reference human model is tetrahedralized into five tetrahedral meshes each of which either can be worked dependently with its adjacent ones or can be worked independently. The clothed reference human model is parametrically encoded in the tetrahedral meshes. Thirdly, these tetrahedral meshes are deformed by fitting the reference human model onto the target human model by using constrained volumetric graph Laplacian deformation. The updated garment models are finally decoded from the deformed tetrahedral meshes. As a result, the updated garment models are fitted onto the target human model. Experiments show that our approach performs very well and has the potential to be used in the garment design industry.

© 2011 Elsevier B.V. All rights reserved.

1. Introduction

Virtual garments are widely used in the industries of garment design, computer generated movies and games. Designing a virtual garment is labor intensive, especially for designing an elegant virtual garment with complex geometry [1,2]. After carefully designing a garment model, it is desirable to have an automatic process that can grade this garment to human bodies with different shapes and sizes [3]. Such a reuse process is very useful. For example, a client likes a garment on the modeler very much, however the garment is not suitable for her/his body, then the garment has to be customized.

A lot of work has been done for garment customization. Among them, the 2D-to-3D schemes of sewing a series of 2D patterns together on a human model to form a garment are prevalent [4–7]. 2D patterns are graded according to individual human body sizes with empirical rules. However, such rules are hard to be implanted into a computational procedure [8]. Before getting a fitted garment, it usually needs several times of interactive modifications on the patterns which are tedious and time consuming.

Our work focuses on customizing 3D garment model directly. We believe that such a 3D customization approach is more intuitive than the 2D-to-3D schemes, since the ultimate aim is to design a 3D garment. For transferring the garment models C initially wore on a reference human model H1 onto a target human model H2, we reuse the human–garment spatial relationship of H1 onto H2, thus fit C onto H2. As shown in Fig. 1, the flow chart of our approach can be decomposed into two parts:

Fitting H1 onto H2: This part sets the spatial mapping from H1 to H2. For achieving this goal, firstly, along each human skeleton, cross-sections are created on each human model. These cross-sections are severed as shape constraints in fitting H1 onto H2 by using Laplacian mesh deformation (LMD). The fitting result H1’ is further mapped onto H2 to get H1” . H1” has the same mesh structure with H1 and is very close to H2 in shape.

Volumetric deformation: This part deforms the shape of C to get C’ which is suitable for H2. For achieving this goal, the clothed H1 is tetrahedralized into five tetrahedral meshes {Ti} (i = 0, 1, 2, 3, 4) each of which either can be worked independently or can be worked dependently with its adjacent ones. H1 and C are encoded in {Ti}. {Ti} is deformed by using volumetric graph Laplacian (VGL) with the constraints extracting from H1 → H1” and other constraints such as skeleton constraints, differential geometry constraints, edge rigidity constraints, etc. C’ is decoded from the deformed {Ti}.

Our work is inspired by Wang et al. [3]. In their work, to transfer a product C wore on a reference model H1 onto another model H2,
they firstly fit $H_1$ onto $H_2$ with using the combination of rigid transformation, radial basis functions (RBF) based elastic warping and iterated closed point (ICP) based method [9]. Then they use compactly supported radial basis functions (CSRBF) to set the volume parameterization from $H_1$ to $H_2$. $C$ is customized simultaneously in this process. However, the detailed methods in our approach are largely different from theirs. Our approach is LMD and VGL based. Comparing with RBF or CSRBF method, the method of LMD or VGL method has following merits:

(a) It is more flexible in integrating various constraints such as position, differential geometry features, and skeleton. Some constraints such as differential geometry features are hard to be used in RBF or CSRBF method, while such constraints are very useful in preserving local geometric details.

(b) LMD or VGL can be implemented by solving linear equations. The coefficient matrix would be constant, if we only update the constraint values while keeping the constraint forms unchanged. This characteristic is very useful in editing the clothed human model. For example, after customizing a garment model onto an individual human model, one may want to check the garment state in different human poses by editing the human skeleton. Such an operation can be efficiently implemented by using VGL, as will be detailed in Section 7. While RBF or CSRBF is implicit function based. They have no such characteristic. When the pose of human model is changed, the coefficients of the implicit function have to be re-computed, which may not be appropriate for interactive clothed human editing.

The rest paper is organized as follows. After reviewing the related work in Section 2, notations for shape encoding/decoding, LMD and VGL are introduced in Section 3. Based on LMD, a new method for fitting one human model onto another is proposed in Section 4. A novel method for tetrahedralizing a clothed human model is provided in Section 5. A human model driven VGL method is provided to transfer garment models on one human model onto another in Section 6. Experimental results and analysis are given in Section 7. The whole paper is concluded in Section 8.

2. Related work

Directly customizing 3D garment models is an emerging technology and the relative literatures are very limited. Before CSRBF based method [3], Wang et al. [10] also proposed an approach for garment customization by “copying” the human–garment spatial relationship from one human model to another. In their work, each garment is represented by a unique feature template which shares common features with the human model. When the shape of human model is altered, the corresponding garment model can be updated automatically. Their approach works well for garments with specified feature templates. However they have not mentioned how to extend their method to the garments without such feature templates such as garment models gotten from commercial software.

Zhong [11] proposed a method for redress a scanned garment model onto individual human models. Rather than customizing the garment models, their method focus on evaluating whether the given garment model is suitable for individual human models or not.

Recently, Li et al. [12] proposed a method for directly fitting 3D garment models onto individual human models. The garment is firstly assigned a skeleton which has similar structure with human skeleton. Then, the garment model together with its skeleton is embedded into a volumetric mesh. The volumetric mesh is deformed by aligning the garment skeleton with the human skeleton, so that the garment model is roughly fit onto the human model. Shape of the garment is further altered by removing the penetrations between the garment model and the human model.

Despite the 3D garment customization, from the view of techniques in detail, our work is relative to shape deformation which is an active filed in computer graphics since the early 1980s. A large variety of approaches have been proposed including freeform deformation (FFD), skeleton based deformation, multi-resolution deformation, RBF-based deformation, differential domain mesh deformation and so on.

FFD and its variants [13–16] deform an object by embedding it inside a volume where the object is parameterized. When the volume is deformed, the embedded object is updated from the volume. FFDs are useful for coarse-scale deformations but not competent for finer-scale deformations.

Multi-resolution techniques [17–19] have been proposed to edit fine meshes. A model is decomposed into a smooth base mesh and several levels of geometric details. The deformed shape is obtained by adding back the geometric details onto the modified base mesh. These methods share a common problem that geometric details are
encoded on each vertex independently which may cause artifacts in highly deformed areas.

Other deformation methods including RBF based [20–22], skeleton based [23] are effective in many applications, however, they have a common limitation as described in FFD and multi-resolution methods that geometry details presented on the mesh surface might be seriously distorted in large scale deformation [24].

Differential methods (refer to [24] for a survey) such as LMD [25,26,36] or Poisson mesh deformation [27,28] rely on local differential geometry, as such, they can preserve local details in surface deformation. However, they can result in a volume loss, if no volumetric constraint is added. Zhou et al. extend the LMD into the volumetric domain [29] to prevent unnatural volume changes. Huang et al. developed this technique further to make it can preserve volumes exactly [30].

Our approach combines VGL with FFD. Shape change of the reference model (human model) is used to control the target model (garment model) indirectly by deforming the volumetric space. The deformation result is achieved by minimizing a constrained Laplacian deformation energy function. A novel method for tetrahedralizing a clothed model is also proposed.

For transferring the garment models on one human model onto another, we need to set the correspondence between the two human models. Mesh cross-parameterizations [31–35] are relative to this problem. By parametrically representing two meshes with a common domain, mesh cross-parameterization provides a bijective map between the two meshes. However to get an appropriate common domain is usually non-trivial.

Human models have intrinsic features such as feature points and skeleton structure which can be used for fitting one human model onto another without explicit parameterization. Allen et al. used feature points together with ICP method to fit one human model onto another [9]. ICP based methods work well for two models with similar shapes and poses. However, when they are in large shape or pose difference, large iteration steps are prone to cause self-intersections, thus it requires many times iterations with appropriate iteration steps before convergence, which is computationally costly.

We also use human features in assistance of fitting one human model onto another. In our approach, we extend the semantic feature from feature points to curves, i.e. cross sections. Curves have more geometric information than points, hence are more effective in surface fitting. Instead of ICP method, we use constrained LMD to fit one human model onto another for efficiency.

3. Background knowledge

3.1. Triangular mesh and tetrahedral mesh

A triangular mesh \( M = (V, E, F) \) is represented by a set of vertices \( V = \{v_i \in \mathbb{R}^3\} \), triangles \( F = \{f_i\} \) and edges \( E = \{e_i\} \), \( f_i = \Delta(v_{i0}, v_{i1}, v_{i2}) \) and \( e_i = (v_{i0}, v_{i1}) \) define the geometry and topology of the surface.

We use bold letter to denote the coordinate vector of a point, e.g. \( \mathbf{v}_i \) for \( v_i \). A point on surface is represented as \( p_i \). Without loss of generality, supposing \( p_i \) is inside face \( f_i \) its barycentric coordinate in \( f_i \) is represented as \((\alpha_i^j, \beta_i^j, \gamma_i^j)\) with

\[
p_i = \alpha_i^0v_{i0} + \beta_i^0v_{i1} + \gamma_i^0v_{i2}, \quad \alpha_i^0 + \beta_i^0 + \gamma_i^0 = 1
\]

This equation holds when \( f_i \) is deformed.

The human model and the garment model are usually triangular meshes. They are represented as \( H \) and \( C \), respectively, \( C \) may contain more than one piece of meshes. In our work, \( C \) is initially dressed on a reference human model \( H_1 \) and is to be customized for a target human model \( H_2 \).

A tetrahedral mesh \( T = (V, E, U) \) is represented by a set of vertices \( V \), edges \( E \), and a set of tetrahedrons \( U = \{u_i\} \), \( u_i = (v_{i0}, v_{i1}, v_{i2}, v_{i3}) \). If a point \( p_i \) is inside \( u_i \), its position \( \mathbf{p}_i \) can be represented as

\[
\mathbf{p}_i = \alpha_i^0v_{i0} + \beta_i^0v_{i1} + \gamma_i^0v_{i2} + \omega_i^0v_{i3}, \quad \alpha_i^0 + \beta_i^0 + \gamma_i^0 + \omega_i^0 = 1
\]

where \((\alpha_i^0, \beta_i^0, \gamma_i^0, \omega_i^0)\) is the barycentric coordinate of \( p_i \) in \( u_i \). This equation holds when \( u_i \) is deformed.

3.2. Encoding and decoding

With Eq. (2), for any spatial point \( p_i \) inside a tetrahedral mesh \( T \), its position can be parametrically represented by a proper tetrahedron \( u_i \) which contains \( p_i \). The following process is named as encoding \( p_i \) in \( T \): finding out the proper \( u_i \) storing \( u_i \)'s pointer and the barycentric coordinate for \( p_i \). Decoding \( p_i \) from \( T \) means updating the position of \( p_i \) from the deformed \( T \) with Eq. (2). Similarly, a point on surface can also be encoded in a proper triangle and decoded from the triangle by using Eq. (1). For a triangular mesh \( M_0 \), encoding its vertices into a tetrahedral mesh \( T \) is named as encoding \( M_0 \), in \( T \), and decoding its vertices from \( T \) is named as decoding \( M_0 \) from \( T \).

With the technique of encoding and decoding, for editing complicated geometric objects, we can encode them into a relative simple tetrahedral mesh, and then indirectly edit them by manipulating the tetrahedral mesh. Such an indirect editing has following merits:

1. The spatial relationship between different objects can be well preserved. That is to say, when we use this technique to edit a clothed human model, the spatial relationship between the human model and the garment model or the spatial relationship between different pieces on a same garment model would be well preserved in the editing results.

2. Manipulating a relative simple tetrahedral mesh would be more efficient and convenient than directly manipulating garment models which usually hold multi-pieces or multi-layers.

3.3. Laplacian deformation on graphs

Our definition on Laplacian deformation follows the literatures [29,36]. Supposing \( G = (V, E) \) is a graph, where \( V \) is a set of vertices and \( E = \{(v_{i}, v_{j})\} \) connects \( v_i \) is the set of edges, the Laplacian of \( G \) computes the difference between each vertex \( v_i \) and a linear combination of \( v_i \)'s neighboring vertices:

\[
\delta_i = L(v_i) = v_i - \sum_{(v_{i}, v_{j}) \in E} w_{ij}v_{j}, \quad \sum_{(v_{i}, v_{j}) \in E} w_{ij} = 1
\]

where \( \delta_i \) is the Laplacian coordinate of the vertex \( v_i \). There are many ways to compute the weight \( w_{ij} \), such as cotangent weight [37], uniform weight [36] and so on. Among them the cotangent weight is the most popularly used in 2D manifold mesh due to its ability to deal with non-uniform tessellations:

\[
w_{ij} \propto \cot \alpha_{ij} + \cot \beta_{ij}
\]

where \( \alpha_{ij} \) and \( \beta_{ij} \) are the two angles opposite to the edge \( v_{i}v_{j} \) in the two triangles linking \( v_{i}v_{j} \). In our work, we use cotangent weight for triangular mesh and use uniform weight for tetrahedral mesh.

When the graph is deformed into a new state with a set of constrained points \( \{p_i, 0 < i \leq m\} \), its deformed position can be
obtained by minimizing the Laplacian deformation energy \( E_L \),
\[
\min(E_L), \quad E_L = \sum \left\| L(v_i') - \delta \right\|^2 + \lambda_1 \sum \left\| p_i - p_i' \right\|^2
\]
where \( p_i \) is the deformed position of \( p \), \( v_i' \) is the deformed \( v_i \), \( \delta_i \) is the Laplacian coordinate of \( v_i' \) and can be computed as
\[
\delta_i' = \mathbf{T}_i \delta_i
\]
where \( \mathbf{T}_i \) is a 3-by-3 transformation matrix which transforms \( v_i \) to its deformed pose \( v_i' \).

4. Fitting one human model onto another

Human model fitting is achieved in three steps. Firstly, cross-sections on each human model are extracted. Secondly, the mapping between cross-sections from different human models is set up. Finally, the reference human model is deformed to the target one with the cross-section shape constraints.

4.1. Cross-section

Cross-sections are curves on a human model surface, which are obtained by intersecting the human model with planes with their normal changed smoothly along the skeleton, as shown in Fig. 2. The detailed method for skeleton and cross-sections extraction is referred to literature [38]. The joint positions can be interactively adjusted to make sure that the joints on the skeleton are located on the positions as the anatomical joint should be.

Cross-sections represent the main geometry of the human models, thus can be served as constraints to set up the mapping between different human models. We control the number of cross-sections on each human model to make that each cross-section \( c_i \) on human model \( H_1 \) has its unique correspondence \( c_i' \) on human model \( H_2 \). We call \( c_i \) and \( c_i' \) as a cross-section pair.

As shown in Fig. 2(a) and (b), the center of \( c_i \) is set to be the intersection point between the human skeleton and the plane containing \( c_i \). For each \( c_i \), a local orthogonal frame \((\mathbf{u}_i, \mathbf{v}_i, \mathbf{w}_i)\) with its center being the center of \( c_i \) is set. \( \mathbf{u}_i \) is the normal direction of the plane containing \( c_i \); \( \mathbf{v}_i \) is the cross production of \( \mathbf{u}_i \) and the view direction (initially, the human models are adjusted to make them face the view direction). The local frame divides a cross-section into four parts, i.e. curves I–IV. On each part, a number of points are equidistantly sampled to set up the bijective mapping between \( c_i \) and \( c_i' \). As shown in Fig. 2(c) where the arrow reflects the mapping from one sampling point on \( c_i \) to \( c_i' \) in part I. Each sampling point on the cross-sections is encoded on the human model by using Eq. (1). These sampling points are served as constrained points in fitting \( H_1 \) onto \( H_2 \), which will be detailed in Section 4.2. The uniformly defined local frames decreases the twisting deformation among neighboring cross-sections, while piecewise mapping \( c_i \) onto \( c_i' \) smoothes the local deformation on \( c_i \).

The interval between every two adjacent cross-sections varies along the skeleton. For the regions holding more geometric details, more cross-sections are distributed, so as to control the human model shape effectively. Totally, 104 cross-sections are created on each human model, as shown in Fig. 2. The interval between every two adjacent sampling points on a cross-section is about 2.8 cm, e.g. 28 points are sampled on a waist cross-section as shown in Fig. 2.

4.2. Cross-sections based human model fitness

Eq. (4) is used to fit human model \( H_1 \) onto human model \( H_2 \). In the equation, the sampling points of the cross-sections on \( H_1 \) are served as constrained points with their objective position superposing their counterparts on \( H_2 \).

For solving Eq. (4), the remaining task is determining the local transformation \( \mathbf{T} \) on each vertex \( v_i \) on \( H_1 \). Similar to [29], in our method, \( \mathbf{T} \) on \( v_i \) is composed with an isotropic scale \( s_i \) and a rotation \( \mathbf{q} \), which is represented as a quaternion, thus \( \mathbf{T} = (s_i \mathbf{q}) \).

For each point sampling \( p_i \) on \( c_i \), its surface normal, \( \mathbf{n}_i \), and cross-section tangent vector, \( \mathbf{t}_i \), construct a local frame \((\mathbf{n}_i, \mathbf{t}_i, \mathbf{r}_i)\) as shown in Fig. 3. Similarly, a local frame \((\mathbf{n}_i', \mathbf{t}_i', \mathbf{r}_i')\) on \( p_i' \), the counterpart of \( p_i \) on \( H_2 \), can be constructed. Then the rotation transformation \( \mathbf{q}_p \) on \( p_i \) can be deduced from the orientation difference between these two local frames. We compute the scale \( s_p \) on \( p_i \) as
\[
s_p = \frac{\mathbf{Y}(\mathbf{c}_i)}{\mathbf{Y}(\mathbf{c}_i')}
\]
where \( \mathbf{Y}(\cdot) \) is a function used to compute the curve length.

With the local transformation on the sampling points \( \{p_i\} \) as constraints, \( \mathbf{T} \) on each vertex \( v_i \) on \( H_1 \) can be obtained with the expectation that local transformation difference among neighboring vertices is as small as possible, which can be achieved by solving a constrained quadric minimization:
\[
\min \left\{ \sum_{(i, j) \in E} \left\| \mathbf{T}_i - \mathbf{T}_j \right\|^2 \right\},
\]
\[
\text{s.t.} \quad \alpha_j T_{i0} + \beta_j T_{j1} + \gamma_j T_{j2} = \mathbf{T}^h
\]
where $T_i$ is the local transformation on vertex $v_i$ which shares an edge with $v_i$. $T^b = (q^b, s^b)$ is the local transformation on sampling point $p_i \in \{p\}; (\alpha_i, \beta_i, \gamma_i)$ is the barycentric coordinate of $p_i$ in its corresponding triangle $f_i$; $T_{0i}, T_{1i}, T_{2i}$ are the local transformation on the three vertices of $f_i$, respectively. $q$ is normalized in the result to get the final $T_i = (q_i, s_i)$.

By substituting the resulting $T_i$ computed from Eq. (6) into Eqs. (5) and (4) and using cross-sections as constraints in Eq. (4), $H_1$ is fitted onto $H_2$. Fig. 4(c) shows the result of fitting $H_1$ shown in Fig. 4(a) onto $H_2$ shown in Fig. 4(b). In Fig. 4(c), overlap between the deformed $H_1$ and $H_2$ reflects the shape difference between them. Fig. 4(d) shows the deformed $H_1$, i.e. $H'_1$.

$H_1$ and $H_2$ may have different scales of surface details. As shown in Fig. 4, hands in $H_1$ have complete fingers, while hands in $H_2$ are in shape of palms. There is no obvious correspondence between these two kinds of hands. If we directly fit hands in $H_1$ onto $H_2$, large deformation will be occurred, which would cause unnatural volumetric deformation in the vicinity of the hands. Consequently the garment parts near the hands would be unnaturally deformed when we transfer the garment from $H_1$ onto $H_2$. For solving this problem, we remove the cross-section constraints on hands and feet when $H_1$ and $H_2$ have largely different scale of surface details. As a result, the pose and shape of hands are transformed self-adaptively as shown in Fig. 4(c) and (d). In experiments, we use 1380 constraint points on the cross-sections in human model fitting, which yields pleased fitting results.

Instead of using points as position constraints in general LMD methods, we use cross-sections as the shape constraints. As two
neighboring cross-sections constrain the shape of a cylindrical surface in-between them, we can control the volumetric change of the cylindrical surface through controlling the shape of the cross-sections. Therefore, cross-sections are served as both surface shape constraints and volumetric constraints. As a result, unnatural volumetric change is reduced in human model fitting.

4.3. Human model fitness refinement

After fitting, the deformed $H_1$ is labeled as $H', H'$ is close to $H_2$, as shown in Fig. 4(c) which however still differs from $H_2$ in local details. The fitness is refined by mapping each vertex $v_i' \in H'$ onto to $H_2$ to get $H\hat{2}$. One possible way is mapping each vertex $v_i' \in H'$ onto to $H_2$ along the surface normal on $v_i'$, however the noise on surface normal may cause undesirable mapping result, as shown in Fig. 4(g) where self-intersection occurs due to the undesirable mapping direction. For reducing the mapping errors, in our work, $H'$ is mapped onto to $H_2$ as follows.

Firstly, for each $v_i' \in H'$, a set of triangles $\{f_i^{H2}\}$ on $H_2$ which are vicinal to $v_i'$ is searched out. Then on each $f_i^{H2}$, compute the point $p_i^{H2}$ which is nearest to $v_i'$. Among $\{p_i^{H2}\}$, the one nearest to $v_i'$ is chosen as the projection point of $v_i'$ on $H_2$, labeling it as $v_i''$. In such a way, the possibility of occurring self-intersection would be greatly reduced. Fig. 4(e) shows the result of mapping $H'$ onto $H_2$, Fig. 4(f) shows $H\hat{2}$ which is the deformed shape of $H'$ after mapping. The shape comparison between $H'$ and $H\hat{2}$ shown in Fig. 4(e) reflects that $H\hat{2}$ fits $H_2$ well. The efficiency of human surface mapping largely depends on the technique of searching out the appropriate $\{f_i^{H2}\}$ for each $v_i' \in H'$. In our work, the efficiency is guaranteed by a pre-segmentation for human models. As shown in Fig. 4(h), each human model is segmented by its skeleton into several regions each of which relates to a bone on the skeleton. In the figure, each region is rendered in a different color. This segmentation technique will be detailed in Section 5. After human model segmentation, the following two-tuples is stored for each vertex $v_i$:

$$(b_j, r_i)$$

where $b_j$ is the index of the bone related to $v_i$, $r_i$ is the length rate of $v_i$’s projection point on the bone. $r_i$ is computed as

$$r_i = \frac{||\vec{v_i} - \vec{j}_1|| \cdot ||\vec{j}_1 - \vec{j}_2||}{||\vec{j}_1 - \vec{j}_2||^2}$$

where $\vec{j}_1$ and $\vec{j}_2$ are the positions of two end points of the bone $b_j$, respectively. “$\cdot$” is dot product of two vector. For each bone, the indices of the triangles each of which has at least one vertex related to this bone is stored.

For each $v_i' \in H'$ with its two-tuples as $(b_j, r_i)$, its candidate mapping triangles $\{f_i^{H2}\}$ are initially chosen as the related triangles of bone $b_j$ on $H_2$. For each $f_i^{H2} \in \{f_i^{H2}\}$, supposing its minimal and maximal bone rate on bone $b_j$ is $r_{\min}$ and $r_{\max}$, respectively. Then among the initial candidate triangles, the triangles satisfying following condition are chosen as the final candidate triangles $\{f_i^{H2}\}$:

$$(\left(r_{\min}^i \leq r_i < r_{\max}^i\right) \land \left(\left|r_{\min}^i - r_i\right| < \delta\right) \land \left(\left|r_{\max} - r_i\right| < \delta\right)) \& \left(\text{norm}\left(f_i^{H2}\right) \cdot \text{norm}(v_i') > 0\right)$$

where “$\land$” and “$\&$” are the Boolean operation of “or” and “and”, respectively. $\text{norm}(f_i^{H2})$ and $\text{norm}(v_i')$ are the surface normal on $f_i^{H2}$ and $v_i'$, respectively. $\delta$ is a threshold. After plenty of experiments, we have found that $\delta = 0.2$ balances the efficiency and accuracy well.

5. Clothed human model tetrahedralization

For a clothed human model $H_c$, its surround space is tetrahedralized into five tetrahedral meshes $(T_i)$ $(i = 0, 1, 2, 3, 4)$. Then $H_c$ is encoded in $(T_0)$ and $(T_i)$ is used to control the shape of $H_c$ in garment customization.

There are a variety of methods for converting surface meshes into volumetric meshes [39–41,29]. These methods work well for closed and manifold surface mesh, however they are difficult in tetrahedralizing unclosed or non-manifold meshes. Garment models are mostly composed with unclosed meshes and sometimes non-manifold meshes. Moreover the ease allowance existing between garment models and human models makes the problem more difficult. To best of our knowledge, there is no tetrahedralization method particular for clothed human models. To address this issue, we propose a new method for tetrahedralizing the space around a clothed human model.

Our tetrahedralization method is inspired from the human structure and garment structure. Anatomically, a human model is formed with arms, legs and torso, while a garment might be composed with sleeves, trouser legs and torso part. Therefore, we can tetrahedralize a clothed human model $H_c$ into five tetrahedral meshes: two arm parts, two leg parts and one torso part. For every two adjacent tetrahedral meshes, they are designed to overlap each other to keep the connectivity in space. Tetrahedralizing $H_c$ in such a way has following merits:

1. The tetrahedralization problem is simplified. Since each sub-part is in cylindrical shape; therefore it is easier to tetrahedralize these parts separately than tetrahedralizing the whole clothed human model directly.
2. Every two adjacent tetrahedral meshes overlap each other, therefore these five tetrahedral meshes either can be worked together as a whole tetrahedral mesh or can be worked independently, which makes it flexible in garment customization operation and in post-processing of clothed human editing as would be detailed in the following content.

5.1. Skeleton based $H_c$ segmentation

Segmenting $H_c$ by its skeleton means that for each $v_i \in H_c$ computing its two-tuples $(b_j, r_i)$. For each vertex $v_i \in H_c$, it is initially bound onto one bone by satisfying the following two criteria:

1. Vertex-bone visibility. For each $v_i \in H_c$ with its normal $\vec{n}$, supposing its projection point on bone $b_j$ is $p_j$ if $(\vec{v}_i - \vec{p}_j) \cdot \vec{n} > 0$, then bone $b_j$ is visible from $v_i$, otherwise bone $b_j$ is invisible from $v_i$.
2. Vertex-bone distance. From all the visible bones from $v_i$, choose the one which is nearest to $v_i$ as the relative bone for $v_i$. Storing the two-tuples $(b_j, r_i)$ for $v_i$.

The above method roughly segments $H_c$ into several regions, as shown in Fig. 5(b) and (c), where different regions are rendered in different colors. However for models which have abundant surface details, such a simple segmentation usually cannot get effective result. As shown in Fig. 5(c), the dress model is not appropriately segmented, especially on the right low hem where different regions are mingled. Such segmentation results are to be optimized by verifying each vertex whether it being appropriately bound to the skeleton or not. Before optimization, the following definition is introduced.

In our work, a human skeleton is composed with 21 bones which are tree structured. For two bones, the bone-bone-distance is defined as the number of bones in-between them. For two vertices
on \(H_c\), the bone-bone-distance between them is defined as the bone-bone-distance between their relative bones. For each vertex \(v_i \in H_c\), its vertex-bone-distance is defined as

\[ d_i = ||f_{ij} + r_i (f_{ik} - f_{ij}) - v_i || \]

where \(r_i\) is the length rate in \((b_p, r_i)\) of \(v_i, f_{ij}, f_{ik}\) are the positions of two joints of bone \(b_p\).

For each \(v_i \in H_c\), among its adjacent vertices, supposing \(v_j\) is the one with smallest vertex-bone-distance. If \(v_j\) satisfies the following conditions, then it is labeled as inappropriately bound:

- (a) \(v_i\) and \(v_j\) are bound onto different bones;
- (b) the vertex-bone-distance on \(v_i\) is larger than that on \(v_j\);
- (c) the bone-bone-distance between \(v_i\) and \(v_j\) is larger than 0.

For a vertex \(v_i \in H_c\) if it is inappropriately bound, its relative \(b_j\) is re-assigned to be the relative bone on its appropriately bound adjacent vertex by using breadth-first algorithm. Furthermore, the small regions are merged into their adjacent larger regions. Fig. 5(d) and (e) shows the optimized segmentation of Fig. 5(b) and (c), respectively.

5.2. Tetrahedral mesh generation

After segmentation, \(H_c\) is decomposed into five sub-parts: two arm parts, two leg parts and one torso part. Each sub-part is extended to overlap one triangle strip on each of its adjacent sub-part. As shown in Fig. 6(a), on each sub-part, the boundary strips rendered with the colors of adjacent sub-parts are the overlapping strips. These overlap strips are used to keep the deformation connectivity across \(H_c\).

Each sub-part is in a cylindrical shape; therefore it can be enclosed with a cylinder-like tetrahedral mesh. In our work, such tetrahedral meshes are created automatically with the strategy as follows.

For each sub-part, a number of planes \(\Pi_i\) intersecting with it are interpolated along the skeleton. For every two adjacent planes, a tetrahedralized cylinder is created to enclose the surface in between them. The final tetrahedral mesh is created by stitching all of such tetrahedralized cylinders together. For the orientation of each \(\Pi_i\), the plane on each skeleton joint bisects the joint angle, while the normal of planes in between two adjacent joints are interpolated smoothly along the skeleton, as shown in Fig. 6(b).

Tetrahedralized cylinder in between each two adjacent planes is created by altering a proper tetrahedralized cylinder template.

Two tetrahedralized cylinder templates are defined, as shown in Fig. 6(c). The upper one is created by cutting a solid cylinder into 8 prisms with \(\angle q_i q_{i+1} q_j = \pi/4\) where \(q_i\) is the center of the upper octagon of the cylinder, \(q_j\) \((j = 0, 1, \ldots, 7)\) is the vertex of this octagon. Each prism is further cut into 3 tetrahedrons, as shown in the right figure in Fig. 6(c). The bottom one is created by cutting a solid cylinder into 8 prisms first and each prism is further cut into 3 smaller prisms, and then these 24 prisms are converted into 72 tetrahedrons. The bottom template is more complex than the above one and is used for the torso part of \(H_c\), while the above one is used for the arm parts and the leg parts of \(H_c\).

The tetrahedralized cylinder between two adjacent planes is obtained by altering the vertex positions of a proper template to make it enclose the \(H_c\) surface in between the two planes. In altering a tetrahedralized cylinder template, we only need to reposition the points of \(o_i\), \(o_{i+1}\), \(q_{i,j}\), \(q_{i,j+1}\) \((j = 0, 1, \ldots, 7)\) as shown in Fig. 6(c), while the other vertices of the deformed template, such as the vertices on the inner octagons of the second template, can be updated with the template definition as described above.

In altering a tetrahedralized cylinder template, its axis \(o_0q_{i,j}\), is set to be parallel with the human skeleton in between \(\Pi_i\) and \(\Pi_{i+1}\); \(o_0q_{i,0}\) and \(o_0q_{i,7}\) are set to be parallel with the axis \(v_j\) and axis \(w_j\) of the local frame of \(\Pi_i\). The definition of such a local frame is given in Section 4.1. The eight planes each of which passing through line \(o_0q_{i,j}\) and \(o_{i+1}q_{i+1,j}\) \((j = 0, 1, \ldots, 7)\) decompose the \(H_c\) surface in between \(\Pi_i\) and \(\Pi_{i+1}\) into eight sectors labeled as \(S_i\) \((j = 0, 1, \ldots, 7)\). The position of \(q_{i,j}\) is re-positioned in such a way that \(S_i\) would be enclosed in the prism \((o_0q_{i,j}, q_{i,j+1}, o_{i+1}, q_{i+1,j+1}, q_{i+1,j+1}, q_{i,j+1})\), as follows.

As shown in Fig. 6(d), passing through each of the following intersection points or vertices inside the sector \(S_i\): the intersection point between the edges on \(H_c\) and the plane \(\Pi_i\) or \(\Pi_{i+1}\), or the vertices on \(H_c\) in between \(\Pi_i\) and \(\Pi_{i+1}\), a plane with its normal being parallelizing with the angular bisector of \(\angle q_{i-1} q_i q_{i+1}\) is used to intersect with line \(o_0q_{i,j}\) line \(o_0q_{i,j+1}\), line \(o_{i+1}q_{i+1,j}\), and line \(o_{i+1}q_{i+1,j+1}\). Among all the intersections on \(o_0q_{i,j}\), the one which is furthest from \(o_i\) is selected as a candidate position \(\hat{q}_{i,j}\) of \(q_{i,j}\), \(\hat{q}_{i,j}\) is updated iteratively by the above treatment for each sector to get the resulting position for \(q_{i,j}\). Fig. 6(d) shows two candidate positions of \(q_{i,j}: \hat{q}_{i,j}^m\) and \(\hat{q}_{i,j}^n\) which are relative to the intersection point \(p_{m,j}\) and vertex \(v_{i,j}\), respectively.

In order to get a good shape of tetrahedralized cylinder, on each plane \(\Pi_i\), the length of each \(o_0q_{i,j}\) is adjusted to be not shorter than the 0.7 times of the maximal length of \(o_0q_{i,j}\) \((j = 0, 1, \ldots, 7)\). Fig. 6(e) shows a top view of the tetrahedralized cylinder in between the second and third intersecting planes on the right arm. In the figure, the red points are the intersection points on the second intersecting plane, and the green points are the vertices of \(H_c\) in between the second and third intersecting planes.
The tetrahedralized cylinders containing the end joints of each sub-part are prolonged along the skeleton, so that every two adjacent tetrahedral meshes overlap each other. Fig. 6(f) shows the tetrahedral meshes for each sub-part. Their relative position is shown in Fig. 6(g) and (h). Fig. 6(h) is rendered in semitransparent. Fig. 6(i) shows the tetrahedral meshes for another clothed human model.

For each sub-part of $H$, its vertices are encoded in its relative tetrahedral mesh. Therefore, each vertex on the overlapped triangle strips is encoded in two tetrahedral meshes. These overlap parts are used to keep the deformation connectivity and smoothness across $H$ in volumetric deformation.

6. Human model driven volumetric deformation

We regard the garments as spatial attachments of human bodies, and assume that the ease allowance between human bodies and garments should be changed smoothly and adaptively according to the shape change of human bodies.

After creating the tetrahedral meshes $\{T_i\}$ ($i=0, 1, 2, 3, 4$) for a clothed human model $H$, with a human model $H_1$ and garment models $C$, $H_1$ is served as shape constraint to deform $\{T_i\}$, so that, when the $H_1$ is altered into $H'_1$, which is gotten in mapping $H_1$ onto $H_2$, $C$ would be transferred onto $H_2$. In deforming $\{T_i\}$, the following deformation energies are expected to be as small as possible.

6.1. Skeleton alignment energy

The skeleton of $H_1$ is deformed to align with that of $H_2$. As the axis of each tetrahedralized cylinder is aligned with a bone segment, for each tetrahedral mesh $T_i \in \{T_i\}$, some of its vertices $\Gamma_i = \{\nu_j\}$ are distributed on the skeleton. For $\nu_j \in \Gamma_i$, its objective position $p_j$ on $H_2$'s skeleton is computed as $p_j = j_f + n(j_f - j_0)$ where $n$ is the length rate in the two-tuples $(b_j, r_i)$ of $j_f$ and $j_0$ are the two end point positions of bone $b_j$. Thus the skeleton alignment energy can be described as

$$E_k = \sum_{i=0}^{4} \sum_{\nu_j \in \Gamma_i} \|v'_j - p_j\|^2$$

where $v'_j$ is the position of $\nu_j \in \Gamma_i$ after deformation.

6.2. Human model fitness energy

In the deformation result of $\{T_i\}$, the final position $v'_j$ of each $\nu_j \in H_1$ can be decoded from the deformed $\{T_i\}$ by using Eq. (2). $v'_j$ is expected to superpose $v''_j$. $v''_j$ is the mapping position of $\nu_j \in H_1$ on $H'_2$. Thus the human model fitness energy $E_{fit}^{H_1}$ of $H_1$ is described as

$$E_{fit}^{H_1} = \sum_{\nu_j \in \Gamma_i} \|v'_j - v''_j\|^2$$

6.3. Laplacian deformation energy on tetrahedral meshes $\{T_i\}$

The Laplacian deformation energy on $\{T_i\}$ can be described as

$$E_L = \sum_{i=0}^{4} \sum_{\nu_j \in T_i} \|L(v'_j) - T_j \partial_j\|^2$$

where $v'_j$ is the deformed vertex of $\nu_j \in T_i$ in the deformation result, $T_j$ is the local transformation on $v_j$. As described in Section 5.2,
vertices in $T_i$ are distributed on a set of planar sections, and on each section, the distribution of the vertices is determined by the local frame of the section. Therefore, $T_i$ can be computed by the local frame based method as described in Section 4.2.

6.4. Laplacian deformation energy on $H_i$

Supposing the shape of $H_i$ decoded from the deformed ($T_i$) is $H_i \setminus H_i^0$, $H_i^0$ is expected to have similar Laplacian attributes as $H_i$. Then the corresponding energy can be described as

$$E_L^{H_i} = \sum_{v_i \in H_i} ||L(v_i^f) - \delta_i^v||^2$$  \hspace{1cm} (10)

where $v_i^f$ is a vertex on $H_i^0$, $\delta_i^v$ is the Laplacian coordinate of $v_i \in H_i^0$. $L(v_i^f)$ can be expressed as a linear combination of the relative vertices on ($T_i$).

6.5. Spatial deformation connectivity energy

($T_i$) is expected to be deformed connectively and smoothly across their overlap parts, so that the resulting $H_i$ decoded from the deformed ($T_i$) is deformed connectively and smoothly. For achieving this goal, two parts of constraints are added. The first part is deduced from the overlapped triangle strips of the 5 sub-parts of $H_i$; the second part is deduced from the overlapped tetrahedrons between two adjacent tetrahedral meshes in ($T_i$).

For the first part of constraints, supposing the vertices on the overlapped triangle strips are $\Gamma_1 = \{v_i\}$, for each $v_i \in \Gamma_1$, it is encoded in two different tetrahedral meshes in ($T_i$). In the deformation result, supposing the position of $v_i$ decoded from the two tetrahedral meshes are $v_i^1$ and $v_i^2$, $v_i^1$ and $v_i^2$ are expect to superpose each other. Thus the following connectivity energy is defined as

$$E_C^{H_i} = \sum_{v_i \in \Gamma_1} ||v_i^1 - v_i^2||^2$$  \hspace{1cm} (11)

where $v_i^1$ and $v_i^2$ are expressed as a linear combination of vertex positions of the two relative tetrahedral meshes, respectively, by using Eq. (2).

For the second part of constraints, for every two adjacent tetrahedral mesh $T_i$ and $T_j$ in ($T_i$), a part of vertices on $T_j$ are inside $T_i$, and vice versa. Supposing $\Gamma_2 = \{v_i\}$ are such vertices. In the deformation result, for each $v_i \in \Gamma_2$, its two candidate position $v_i^1$ and $v_i^2$ computed from the two different tetrahedral meshes are expected to superpose each other. Thus the corresponding connectivity energy is defined as

$$E_C^T = \sum_{v_i \in \Gamma_2} ||v_i^1 - v_i^2||^2$$  \hspace{1cm} (12)

$\Gamma_2$ could be selective for different kind of garment, which would be discussed in Section 7.

6.6. Edge rigidity energy

For each edge $e_i \in H_i$, if its two endpoints are encoded in a same tetrahedron, in the deformation result, $e_i$ would still inside the tetrahedron since a tetrahedron is a convex polyhedron. However, if $e_i$ intersects more than one tetrahedron, i.e. the two endpoints of $e_i$ are encoded in two different tetrahedrons, e.g. edge $v_i^{1f}v_i^{2f}$ in Fig. 7(a), in the deformation result, part of $e_i$ might be outside the corresponding tetrahedrons, e.g. edge $v_i^{1f}v_i^{2f}$ in Fig. 7(b), which however is not a desirable result.

For each $e_i \in H_i$, if $e_i$ holds one or more intersection points with the tetrahedrons along $e_i$. Each intersection point could be either interpolated by three vertices on a proper triangle or interpolated by the two end points of $e_i$. However, coordinates computed from these two channels are usually not converged, as shown the intersection point $(p_i^{00})$ computed from edge interpolation and $(p_i^{01})$ computed from triangle intersection in Fig. 7(b). Actually, $v_i^{1f}$/$(p_i^{0i})$,$v_i^{2f}$ becomes a curve while not a line segment it should be. For address this issue, these two candidate coordinates are constrained to superpose each other, thus to increase the edge rigidity. Hence, we have following edge energy

$$E_E^0 = \sum_i \sum_{m} ||(p_{m}^{0i}) - (p_{m}^{1i})||^2$$  \hspace{1cm} (13)

where $(p_{m}^{0i})$ and $(p_{m}^{1i})$ are the two candidate coordinate for the intersection point $p_{m}$, For $p_{m}$, the superscript "i" means the intersection point is on $e_i$, and the subscript "m" means it is the mth intersection point on $e_i$.

6.7. Summarized energy

Eqs. (7)–(13) are functions of the deformed vertex positions of ($T_i$). Therefore, the resulting shape of ($T_i$) can be obtained by minimizing the linear combination of the above energies as

$$\min(E) \quad E = \lambda_1 E_L^H + \lambda_2 E_C^H + \lambda_3 E_C^T + \lambda_4 (E_E^0 + E_E^1) + \lambda_6 E_C^H$$  \hspace{1cm} (14)

where the coefficients $\lambda_i$ ($i = 1, 2, \ldots, 6$) is used to balance the importance of the six terms. Comparatively, the skeleton alignment and human model fitness are rigid constraints. After a plenty of experiments, we have found the following parameters setting is appropriate: $\lambda_1 = \lambda_2 = 10, \lambda_3 = \lambda_5 = \lambda_6 = 3, \lambda_4 = 1$.

In our method, the volume of ($T_i$) changes according to the shape difference between $H_i$ and $H_{i+1}$, hence, the human–garment ease allowance of $H_i$ is adjusted in the deformed ($T_i$). That is to say, the garment models are customized with the human–garment ease allowance adjusted self-adaptively according to the shape difference between the reference human model and the target human model, which makes the customized garment be suitable for the target human model.

7. Experimental results and analysis

Four human models and three sets of garment models are used for testing the performance of our approach, as shown in Figs. 8–10. In the figures, garment models initially dressed on the human models colored in golden yellow are to be customized to fit the human models colored in orange red. For clarity, the human models used in Fig. 8(a)–(d) is labeled as $H_0$, $H_1$, $H_2$, and $H_3$, respectively. The number of triangles and vertices on each human model is listed in Table 1. These human models vary in shapes and poses. These garment models have different scales of shape details, e.g. the garment models in Fig. 8 are composed with 24
Fig. 8. Example 1: transferring garments on $H_A$ to $H_B$, $H_C$ and $H_D$, respectively. (a) Garments on $H_A$. (b) Customized garments on $H_B$. (c) Customized garments on $H_C$. (d) Customized garments on $H_D$. 

components. For visualization, different components are displayed in different colors. As shown in the figures, our approach can keep the relative position of the components well. (For interpretation of the references to color in this paragraph, the reader is referred to the web version of the article.)

Tetrahedral meshes are created for each reference clothed human model for garment customization. The number of vertices and edges of each tetrahedral mesh are listed in Table 2. Garment customization is achieved by solving Eq. (14). As mentioned in Section 6, in Eq. (14), the energy item $E_C'$ could be...
selective according to what kind of garment models are to be customized. For customizing trousers, such energy item for the two leg parts is removed from Eq. (14), so that the two leg parts would be deformed comparatively independently, as shown in Fig. 8. If such energy item is included in Eq. (14), and if the two tetrahedral meshes overlap each other in the initial state, then deformation on one leg part would affect the another one, as a result, unnatural deformation would be occurred, as shown in Fig. 11. However, in customizing a skirt model as shown in Fig. 9, Such energy item should be used to keep the deformation being smooth across the surface of skirt; otherwise, unnatural results such as self-intersection or over stretching would be occurred, as shown in Fig. 12.

Eq. (14) can be converted into a linear equation system by setting the gradient of $E$ with respective to each new vertex position to be zero, as

$$\mathbf{LX} = \mathbf{B}$$

where $\mathbf{L}$ is the coefficient matrix, $\mathbf{X}$ is the vector of unknown new vertex positions in deformed $\{T_i\}$, $\mathbf{B}$ is the vector of known value from
constraints. We use CHOLMOD [42] to solve Eq. (15). The computational time is listed in Table 3. Our experiments ran on a computer with Intel(R) Core(TM) i7 CPU 920 @ 2.67 GHz + 4.0 GB RAM. From Table 3, we have the following conclusion: the computational time is mainly relevant to the geometric complexity of $H_1$, while is less relevant to the geometric complexity of $H_2$ and $C$.

Our method can geometrically simulate the clothed human models. As shown in Fig. 13, a clothed woman model is driven by its skeleton. For achieving this, in Eq. (14), $\lambda_2$ is set to be zero and $\hat{E}_C$ for the two leg parts is removed. In this processing, $L$ in Eq. (15) is constant, so is $(L^T L)^{-1}$. Thus the vertex positions of $\{T_i\}$ can be efficiently computed as $X = (L^T L)^{-1} B$, e.g. the computational time for each frame in Fig. 13 is about 0.15 s, which well supports the interactive editing.

Though our method can geometrically customize garment models, and produces visually plausible results, it lacks metrics to measure the quality of customization. To address this issue, in the future, we would like to integrate our method with physics based cloth simulation [43]. We would use our geometric method to efficiently deform the clothed human first, and then use physical method to augment the reality.

Our method is divide-and-conquer. A clothed human model is controlled by five tetrahedral meshes, which provides a possible

![Fig. 10. Example III: transferring garments on $H_B$ to $H_A$, $H_C$, and $H_D$, respectively.](image1.png)

![Fig. 11. Keeping $E^H_2$ for the two leg parts in Eq. (14) would cause unnatural deformation on the trousers.](image2.png)

### Table 1
Human mesh information.

<table>
<thead>
<tr>
<th></th>
<th>$H_A$</th>
<th>$H_B$</th>
<th>$H_C$</th>
<th>$H_D$</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Faces</td>
<td>13,598</td>
<td>38,564</td>
<td>11,996</td>
<td>31,446</td>
</tr>
<tr>
<td>#Vertices</td>
<td>6799</td>
<td>19,284</td>
<td>6000</td>
<td>15,755</td>
</tr>
</tbody>
</table>

### Table 2
Tetrahedral mesh information.

<table>
<thead>
<tr>
<th></th>
<th>Tetrahedral mesh of torso part</th>
<th>Tetrahedral mesh of left arm part</th>
<th>Tetrahedral mesh of right arm part</th>
<th>Tetrahedral mesh of left leg part</th>
<th>Tetrahedral mesh of right leg part</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Vertex</td>
<td>425</td>
<td>108</td>
<td>108</td>
<td>153</td>
<td>153</td>
</tr>
<tr>
<td>#Edge</td>
<td>2925</td>
<td>467</td>
<td>467</td>
<td>672</td>
<td>672</td>
</tr>
</tbody>
</table>

### Table 3
Computational time.

<table>
<thead>
<tr>
<th>Fig.</th>
<th>$H_1$</th>
<th>$H_2$</th>
<th>$C$ component no.</th>
<th>$C$ face no.</th>
<th>$C$ vertex no.</th>
<th>Human model fitting time (s)</th>
<th>Volumetric deformation time (s)</th>
<th>Total computational time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 8</td>
<td>$H_A$</td>
<td>$H_B$</td>
<td>24</td>
<td>18,978</td>
<td>9804</td>
<td>4.8</td>
<td>4.2</td>
<td>9.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>4.5</td>
<td>4.1</td>
<td>8.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>5.2</td>
<td>4.0</td>
<td>9.2</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>$H_A$</td>
<td>$H_B$</td>
<td>7</td>
<td>5783</td>
<td>3256</td>
<td>7.3</td>
<td>6.9</td>
<td>14.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.1</td>
<td>6.5</td>
<td>13.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.8</td>
<td>6.4</td>
<td>14.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.3</td>
<td>6.8</td>
<td>14.1</td>
</tr>
<tr>
<td>Fig. 10</td>
<td>$H_A$</td>
<td>$H_B$</td>
<td>1</td>
<td>3953</td>
<td>2060</td>
<td>7.1</td>
<td>6.6</td>
<td>13.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>7.8</td>
<td>6.3</td>
<td>14.1</td>
</tr>
</tbody>
</table>
way for efficient collision detection in garment customization and
clothed human simulation. For detecting the collision between
different parts of a clothed human model, we can detect the
collision between the relative tetrahedral meshes first. The
collision regions of the clothed human are within the tetrahedrons
where collision occurs, which can reduce the computation,
especially when the garment models are complex and the number
of triangles on the clothed human model is large. We would
develop such collision detection method in the future work.

8. Conclusions

In this paper, we have proposed a novel framework for
customizing 3D garment models. The framework is composed
with three main techniques:

(1) Human model fitting, which sets shape-mapping from a
reference human model to a target one. Unlike existing
methods most of which are ICP based with feature points as
shape constraints, our approach uses Laplacian deformation
method with cross-sections as shape constraints. Other than
severing as position constraints as feature points, cross-
sections can also sever as volume constraints which is helpful
in reducing the unnatural volume change in human model
fitting.

(2) Clothed human model tetrahedralization, which tetrahedralizes
the space around a clothed human. Unlike existing methods
most of which convert a closed shell mesh into a tetrahedral
mesh, our approach converts the space around a clothed human
model into a set of tetrahedral meshes. These
tetrahedral meshes either can be used independently to
control a proper part of the clothed human, or can be used
together to control the whole clothed human model.

(3) Human model driven volumetric Laplacian deformation, which
deforms the tetrahedral meshes with the constraints extracted
from human model fitting. As a result, the garment models

Fig. 12. Self-intersection or over stretching would be occurred on the customized skirt, if $E_L$ for the two leg parts is excluded from Eq. (14).

Fig. 13. Driving a clothed human model by its human skeleton.
initially dressed on the reference human model are fitted onto the target human model.

Our techniques can benefit the garment design industry. When a designer has carefully designed a garment on a reference human model, she/he can use the software tool implanting our techniques to efficiently customize the garment, thus the reusability of the design results has been greatly improved. As a result, the design cost and design period can be reduced.

Our techniques are useful for online shop store system. When a client favors the dress on a modeler, before buy, she/he can use our technology to customize the dress and evaluate the dressing effect, as long as she/he provides her/his 3D body model. Such individual models can be gotten by 3D scanning or by the parameterization technologies [9,38,44,45].

Acknowledgements

This work has been supported by National Natural Science Foundation of China (NSFC, No. 60903145), Beijing Natural Science Foundation (4102062). We would like to thank Digital Fashion Ltd. and Cyberware Ltd. for providing the 3D models.

References