Fitting 3D garment models onto individual human models

Jituo Li a, b, c, *, 1, Juntao Ye a, Yangsheng Wang a, Li Bai d, Guodong Lu c

a Digital Interactive Media Lab., Institute of Automation, Chinese Academy of Sciences, No. 95, Zhongguancun East Road, Beijing 100190, PR China
b Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University, Shanghai 201620, PR China
c Institute of Engineering and Computer Graphics, Mechanical Department of Zhejiang University, Room 416, No. 1 Teaching Building, Zhejiang University Yuquan Campus, No. 38, Zheda Road, Hangzhou 310027, PR China
d School of Computer Science, University of Nottingham, Nottingham, UK

A B S T R A C T

Designing an elegant 3D virtual garment model for a 3D virtual human model is labor-intensive, because most existing garment models are custom-made for a specific human model and cannot be easily reused for other individuals. In this paper, we propose a novel method for fitting a given 3D garment model onto human models of various body shapes and poses. The fitting is accomplished by deforming the garment mesh to match the shapes of the human models by using a combination of the following: skeleton-driven volumetric deformation, garment–human shape similarity matching and evaluation, the constraints of garment–human contact, and garment–human ease allowance. Experiments show that our approach performs very well and has the potential to be used in the garment design industry.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

The topic of 3D virtual garment model design has attracted keen interest from the computer-aided design (CAD) community [1–3]; additionally, it is also crucial to the development of high-quality computer generated movies and video games, since a proper garment model can be instrumental in effectively shaping characters and facilitating storytelling. Accordingly, recent developments in clothing simulation research have made it possible to synthesize virtual garments with dynamic behaviors [1,2]. Unfortunately, designing an elegant 3D virtual garment model is labor-intensive, and is not an easy task for non-professionals. A traditional way of building a garment mesh is to use 3D modelers such as Maya and 3DS Max [4] although new technologies based on sketching have emerged; the latter method relies on curves drawn in multi-view directions to construct the 3D garment models [3,5]. Unfortunately, though sketch-based methods are efficient in the modeling of simple garments, they become too complex to create more complicated garment models. Clearly, a better method is needed.

An alternative way to build a garment model is based on the 2D-to-3D scheme. In this method, a series of 2D patterns are first created and exported from pattern design tools, such as Geber [6], Lectra [7], and Investronica [8]. These patterns are then sewn together in a 3D scene to form clothing on a human model. Physics-based techniques are usually adopted to assist the deformation of the 2D patterns into a 3D mesh. Unfortunately, it has been shown that it takes years for a person to become an expert pattern-maker; even the design of simple garments, such as T-shirts and skirts, requires the expertise of an experienced pattern maker [9]. Therefore, as an alternative to fitting 2D patterns onto 3D characters using physics-based method, Igarashi and Hughes [10] proposed a purely geometrical method to wrap 2D clothes onto 3D characters. In their method, the user is asked to draw free-form marks on the 2D clothes and corresponding marks on the 3D character; the system then wraps the clothes onto the character so that the marks match. Our approach is also geometrically based but it does not deal with 2D clothes. Instead, it fits 3D garment models onto 3D human models directly, retaining the global and local details of the garment model as much as possible.

Since a garment model is often initially customized for the particular shape of a specific human body, it will not generically fit other bodies with different shapes; for this reason, garment models are therefore not only expensive to build, but also hard to reuse. In many cases, it is desirable to have garment models of the
same style, but of differing sizes, so they can fit a variety of body meshes. One instance of this technique occurs in computer-aided garment design, or online garment store systems, where a template cloth mesh is fitted to various input body meshes; such technologies, which improve the reusability of garment models, are in high demand by industry, and therefore comprise the focus of our research.

Unfortunately, current technologies for garment model reuse are very limited. While it is possible to use the above-mentioned 2D-to-3D design software to map 2D patterns of differing sizes to various body dimensions, the process is as work-intensive as building entirely new meshes from scratch, thus negating the intended benefit of reuse. Clearly, the process needs modification.

Accordingly, some researchers have attempted to use geometric parameterization methods [11–13] to improve the reusability of 3D garment models. This method uses a human reference model, another (target) human model, and the many pre-defined, corresponding feature points between them, to project the spatial relationship between the garment and reference models onto the target human model; thus, the garment model is deformed to fit the target human model.

The methods provided in [11–13] work well when a human reference model is used in conjunction with the garment model; however, they are not suitable for customizing garments in the absence of the initial reference human models. Nevertheless, such garments can still be created by emerging technology methods such as cut-and-paste based modeling; this method eschews the use of reference models entirely, relying solely on the reuse of existing models [14,15]. Highly efficient methods such as this might alleviate the dependence on human reference models.

Like the cut-and-paste approach, our method does not require the use of a human reference model. As shown in Fig. 1, our method aspires to fit a given 3D garment model onto a human model of arbitrary shape, size, and pose, using the following three novel approaches:

1. Unlike existing garment fitting methods that sew 2D patterns on 3D human models, our method directly fits the 3D garment models onto 3D human models. To the best of our knowledge, there has been no literature addressing such method so far.

2. We convert industrial garment design knowledge into geometric constraints, including the loading region, the fashion region, garment–human contact, and garment–human ease allowance; this enhances garment fitting and yields more realistic results.

3. A series of methods, such as skeleton-driven volumetric deformation, garment–human shape similarity matching and evaluation, garment–human penetration resolution and garment–human ease allowance adjustment, has been incorporated.

2. Related work

Fitting 3D garments onto human models is similar to shape registration. The goal of shape registration is finding a rigid transform between two given shapes that optimally positions one shape against the other. Generally, shape registration methods fall into two classes [16]. The first class, known as the voting methods, exhaustively searches for the transform parameters [17,18]; this method usually can get optimal results, but the computational cost is very high. Alternatively, the second class is based on shape matching, where each point on one shape is correlated to the closest point in another shape. Accordingly, given a set of point pairs, a rigid transform can be obtained using the iterated closest point (ICP) method [19–21]. Irrespective of the selected class, the overall performance of shape registration can be improved using local geometric descriptors; these descriptors can be curvature and its variants [22,23], spin images [24], shape contexts [25,26], integral volume description [27], multi-scale features [28], and so on. In our method, the local descriptors include Euclidean distance and surface normal, which set the correspondence between the garment and the human models. Moreover, we have also proposed a mechanism to evaluate the accuracy of the matched results; instead of using rigid transformation, shape matching for non-rigid garment deformation is used.

Free-form deformation (FFD) and its variants [29,30] play an important role in geometric modeling and editing. In FFD, the free-form object to be deformed is embedded inside a volume that is usually parametrically represented; when this volume is subsequently deformed, the embedded object assumes a new shape. FFD is useful for coarse-scale deformation, but not adequate for fine-scale deformation.

The skeleton-based deformation methods are widely used on articulated models for pose-editing or for animation. The traditional skeleton-based methods are generally based on vertex weighting or linear skin blending [31,32]. Each vertex is controlled by one or more bones with appropriate weights; when the skeleton deforms, each vertex is updated independently. These methods are prone to mesh self-intersections near the joints; therefore, careful weight tuning is required to avoid visual artifacts [33,34].

Sumner et al. [35] have proposed a space deformation method consisting of a collection of affine transformations organized in a graph structure. This method has drawbacks similar to that of the skeleton-based deformation method: because it uses Euclidean distance for its deformation function, artifacts can result while deforming a model that has parts that are too close in Euclidean distance, but too far apart in geodesic distance.

In contrast to the above Cartesian coordinates based methods, the Laplacian mesh deformation (LMD) methods [36,37] and Poisson shape editing methods [38] rely on local differential geometry; as
such, they can preserve local details well during surface deformation. General LMD methods deal with shell models, which are, nevertheless, less appropriate since they can lead to a loss of the volume. To address this issue, Zhou et al. [39] extend the LMD into the volumetric domain; as such, their method can prevent unnatural volume changes and avoid local self-intersections in the deformation results. Huang et al. [40] further developed this technique so that volumes are preserved precisely.

Cage-based deformation is an emerging technique that has proven to be very powerful in space deformation. A cage, typically a low polygon-count closed polyhedron, represents the silhouette of the input model. A spatial point inside the cage can then be represented as a linear combination of the vertices of the cage multiplied by barycentric coordinates; barycentric coordinates can be mean-value [40,41], harmonic [42,43], or Green coordinates [44]. A target cage can be obtained by altering the position of the input model. A spatial point inside the cage can then be represented by its barycentric coordinate \((w_0, w_1, w_2)\) on \(f_j\) as

\[
p_i = w_0v_{i,0} + w_1v_{i,1} + w_2v_{i,2}, \quad w_0 + w_1 + w_2 = 1, \quad 0 \leq w_i \leq 1
\]

The human model and the garment model are represented as \(H\) and \(G\), respectively. The vertices on \(G\) are represented as \(V^G = \{(v_j)\}\) whose matching pairs on \(H\) are represented as \(P^H = \{(p_i)\}\). The vertices on \(G\) are represented as \(V^G = \{(v_j)\}\) whose matching pairs on \(H\) are represented as \(P^H = \{(p_i)\}\). The number of vertices on \(G\) and \(H\) is \(N^G\) and \(N^H\), respectively.

A tetrahedral mesh \(T = (V,E,U)\) is represented by a set of vertices \(V\), a set of edges \(E\), and a set of tetrahedrons \(U = \{u_i\}\), where \(u_i = (v_{i,0}, v_{i,1}, v_{i,2}, v_{i,3})\). If a point \(p_i\) is inside \(u_i\), its position \(p_i\) can be represented by its barycentric coordinate \((w_0, w_1, w_2, w_3)\) in \(u_i\) as

\[
p_i = w_0v_{i,0} + w_1v_{i,1} + w_2v_{i,2} + w_3v_{i,3}, \quad w_0 + w_1 + w_2 + w_3 = 1, \quad 0 \leq w_i \leq 1
\]

### 3.1. Notations on mesh

A mesh \(M = (V,E,F)\) is represented by a set of vertices \(V = \{v_i \in \mathbb{R}^3\}\), a set of triangles \(F = \{f_i\}\), and a set of edges \(E = \{e_i\}\), where \(f_i = (v_{i,0}, v_{i,1}, v_{i,2})\) and \(e_i = (v_{i,0}, v_{i,1})\) define the geometry and topology of the surface. The coordinate of \(v_i\) is represented as a vector, \(v_i\). A point on surface is represented as \(p_i\) and its coordinate as a vector, \(p_i\). Without loss of generality, let us assume that \(p_i\) is inside triangle \(f_i\); its position can be represented by its barycentric coordinate \((w_0, w_1, w_2)\) on \(f_i\) as

\[
p_i = w_0v_{i,0} + w_1v_{i,1} + w_2v_{i,2}, \quad w_0 + w_1 + w_2 = 1, \quad 0 \leq w_i \leq 1
\]

### 3.2. Laplacian deformation on graphs

Suppose \(\text{Graph} = (V, E)\) 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 a graph computes the difference between each point \(v_i\) and a linear combination of its neighboring points:

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

where \(\delta_i\) is the Laplacian coordinate of the vertex \(v_i\). For a triangular mesh, weights can be chosen as cotangent functions

\[
w_{ij} \propto \cot \theta_i + \cot \theta_j
\]

where \(\theta_i\) and \(\theta_j\) are the two angles opposite to the edge \(v_i, v_j\) in the two triangles sharing this edge. For a tetrahedral mesh, uniform weight is used

\[
w_{ij} = 1/n_i
\]

where \(n_i\) is the number of adjacent vertices of \(v_i\).

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, \(E_L\)

\[
\min(E_L), \quad E_L = \sum_{i=0}^{N} \|L(v_i) - \delta_i\|^2 + \lambda \sum_{i=0}^{m} \|p_i - p_i\|^2
\]

where \(N\) is the number of vertices on the graph, and the weight \(\lambda\) indicates the importance of \(p_i\) in position constraints; \(p_i\) is the target position of \(p_i\) while \(v_i\) is the deformed vertex. Additionally, \(\delta_i\) is the Laplacian coordinate of \(v_i\), and can be computed as follows:

\[
\delta_i = T_i\delta_i
\]

where \(\delta_i\) is the Laplacian coordinate in the rest pose, and \(T_i\) is a 3-by-3 transformation matrix, which transforms \(v_i\) to its deformed pose, \(v_i'\).

### 4. Aligning garment model with individual human models

Generally, a garment assumes the shape of human body it covers. Therefore, garment models can be treated as articulated models towards human models. For pose adjustment, skeleton-based deformation is the most intuitive and an easy-to-operate method among a large variety of deformation methods. In this section, we describe how a skeleton-driven volumetric deformation
is used to roughly align the garment model with the target human model.

4.1. Skeleton creation, adjustment, and embedment

A plethora of automatic methods [46,47] exist for creating garment skeletons. However, due to the large variety of garment styles, it is hard to create a universal skeleton with constant structure applicable to different kinds of garment models. Instead of using those automatic methods, we prefer to start with a duplicated human skeleton and manipulate to make it match the shape of the garment model. This skeleton is then embedded inside the garment model as the garment skeleton. Once the garment skeleton is constructed, it is reversely transformed to assume the pose of the human skeleton, driving the deformation of the enclosing garment mesh via our skeleton-driven volumetric method. As a result, the garment pose will be aligned with the human model. For the human skeleton, it can be automatically created by either using our previous work [48] or can be manually created by commercial software such as Maya or 3Ds Max[4].

With a copy of the human skeleton as the starting point, the user is asked to drag each joint to a new position, so that its on-screen projection coincides with that of the corresponding garment joints. While a joint is being dragged, its movement is constrained to be within the plane perpendicular to the view direction. If any intersections between the line and the garment model exist, $f_c^G$ is referred to as a constrained joint and its position is updated as the average position of all intersections. Alternatively, if no intersections exist, $f_c^G$ is referred to as a free joint. When both constrained and free joints are present, a free joint can be repositioned by its two adjacent constrained joints. As shown in Fig. 2(c), where $J_i^G$ and $J_k^G$ are the adjacent joints of $J_j^G$, and $J_j^G$ is a free joint while $J_i^G$ and $J_k^G$ are constrained joints, the position of $J_j^G$ is updated as follows:

$$J_j^G = s_j k R_k (J_i^G - J_k^G) + J_j^G,$$

where $J_i^G, J_k^G$, and $J_j^G$ are the positions of garment joints $I_i^G, J_k^G, J_j^G$, respectively; $J_i^H, J_k^H, J_j^H$ are the positions of human joints $I_i^H, J_k^H, J_j^H$, respectively, and $R_k$ is a 3-by-3 rotation matrix, which rotates the vector $J_i^H - J_k^H$ to the vector $J_i^G - J_k^G$.

Once a free joint is repositioned, it is then set as a constrained joint. All free joints are repositioned using the above method recursively. Therefore, if there exist two or more constrained joints, all the free joints can be repositioned automatically. Nevertheless, if needed, any garment joint can be repositioned by dragging it under different views directions while constraining its motion inside the plane being perpendicular to the view direction.

It must be noted that the garment skeleton does not need to match the garment surface accurately, since the goal of skeleton-driven volumetric deformation is to roughly align the garment model with the target human model. This is because the alignment will be further adjusted with local constraints (Section 5). For the same reason, the volume does not have to be accurately recorded during volumetric deformation as in Huang's method [40], which requires the resolution of non-linear equations. For computational efficiency, we instead rely on linear system of equations, first computing $T_i$ from Eq. (5), then solving Eq. (4) separately, and finally computing the minimization of $E_2$ in Eq. (4); this procedure will be discussed in more detail in the following section.

4.2. Skeleton-driven volumetric deformation

To reduce the distortion of the garment model, the skeleton-driven volumetric deformation method is adopted for adjusting the garment pose.

A set of methods can be used to convert triangular meshes into volumetric meshes [39,49–51]; these tetrahedralizing methods work well for closed meshes, but not for open meshes, such as garment models. We propose a method for searching and tetrahedralizing the approximate bounding volume of the garment model $G$ to get a tetrahedral mesh $T$ as follows:

(1) For each boundary loop of $G$, compute its minimum convex hull. The boundary loop is composed by the continuously connected boundary edges. These convex hulls, together with
G, compose a closed space that can be considered as an approximate bounding volume of G.

(2) Voxelize the bounding box of the convex hulls and G. Flag grids intersecting with any convex hull or G as valid grids.

(3) For each boundary grid g_i, if it is not a valid grid, set it as invalid. For each invalid grid, flag its unflagged adjacent grids as invalid grids recursively. Flag the remaining unflagged grids as valid grids.

(4) Tetrahedralize valid grids to get the tetrahedral mesh T. Set up a connection between the neighboring tetrahedrons.

(5) Encode the position of each vertex on G in T using Eq. (2).

To deform T with the skeleton, we need to first segment T to decide which bone should be used to control each vertex. At each joint where bones meet, we use a plane bisecting the joint angle to segment T. Thus, each vertex on G is correlated to an appropriate bone on C's skeleton. Fig. 2 illustrates the deformation of a garment with a skeleton. Fig. 2(d) shows the corresponding tetrahedral mesh T of the garment model of Fig. 2(b). In the figure, we use different colors to indicate the segmentation result.

To deform T with Eq. (4), we need to specify the position constraints in Eq. (4) and the rotation matrix \( T_i \) in Eq. (5). The position constraints are obtained from the garment skeleton: each joint \( J_i \) of the garment skeleton is mapped onto the corresponding joint \( J_i^h \) of the human skeleton. In addition to the joint constraints, a set of sample points along each bone of the garment skeleton are automatically distributed; the distance between two adjacent samples equals to the average edge length of G. The joints and the sample points are encoded in T with Eq. (2); we then map these joints and samples onto the human skeleton by length mapping.

The length mapping occurs as follows. Let \( Q_i \) be a quaternion representing the rotation from bone \( B^G_i \) on the garment skeleton to its corresponding bone \( B^H_i \) on the human skeleton. \( Q_i \) is set to be the rotation on the midpoint of bone \( B^G_i \). The rotation on each joint is set as the average rotation of its adjacent bones. The rotation of a random \( p_i \) inside \( B^G_i \) is computed as the spherical linear interpolation between the rotation on the midpoint of \( B^G_i \) and the rotation on one end point of \( B^G_i \). Then, for each vertex \( v_i^G \) on G, its corresponding \( T_i \) is set in Eq. (5) as the rotation of \( v_i^G \)'s nearest point on \( v_i^H \)'s corresponding bone. In this way, the local rotations on the garment surface change smoothly and local self-intersections can be avoided, as shown in Fig. 2(g).

Fig. 2(e) shows the deformation result when aligning the garment model with the human model. Our skinning method differs from general skinning methods as follows:

(1) Our skinning method is volume based. The garment model and its related skeleton are parametrically represented by one tetrahedral mesh. Unnatural volumetric deformation and local self-intersections are avoided; thus the embedded garment model is deformed smoothly. While for other skinning methods, the position of each vertex is controlled separately by one or a few bones, unaware of the positions of other vertices, so local self-intersections are prone to occur around the joints in bending the skeleton. Moreover, unnatural volumetric deformation occurs even more frequently.

(2) Our skinning method is efficient and is very easy to implement. The skinning process can be accomplished automatically. Whereas in the general skinning methods, interactive weight tuning is usually required to get an appropriate skinning result, and such tuning work can only be done by experts.

However, we consider that the tetrahedral mesh performs better than other volumetric meshes. In our experiment, a tetrahedral mesh is constructed from a voxel mesh by subdividing each cube into six tetrahedrons. By doing so, the number of mesh edges increases, but the number of mesh vertices is unchanged. A tetrahedral mesh is stiffer than the voxel mesh under shearing force; therefore, it performs better in preventing unnatural volumetric deformation than voxel mesh. Furthermore, since the number of vertices does not change while converting a voxel mesh to a tetrahedral mesh, the coefficient matrices in Eq. (4) have the same valence, independent of the mesh type. Therefore, we consider processing a tetrahedral mesh is as efficient as processing a voxel mesh, while the former gives better quality in volumetric deformation. As shown in Fig. 2(e) and (f), the garment model is deformed more smoothly by using tetrahedral mesh than by using voxel mesh.

5. Adjusting garment shape with local constraints

After garment pose adjustment, the garment model fits the human model approximately. However, it is still not satisfactory, because, in some regions, penetration between the garment and the human body occurs, and in others, the ease allowance between the garment and the human body is either too large or too small. We resolve these problems by local adjustment of the garment shape.

Instead of using a Laplacian volumetric deformation method for local adjustment, we now adopt the Laplacian mesh deformation method, because at this stage we want the garment surface to match the human model perfectly; volume change is therefore necessary due to the shape difference between the garment model and the human model.

Since the garment pose has been already been aligned with the human pose in the global alignment stage, the local rotation of each vertex would be very small in this local adjustment stage. To this end, for computational efficiency, it is reasonable to set \( T_i \) in Eq. (5) as an identity matrix.

5.1. Loading region and fashion region

A garment can be partitioned into many regions, which fall into two general categories—the loading regions (otherwise known as fitting regions) and the fashion regions [52]. A loading region is defined as being in direct contact with the body surface; it is usually designed to counter the gravity of garments and is responsible for garment comfort. A fashion region, on the other hand, is usually not in direct contact with the body, but is draped freely to enhance aesthetic appearance.

The loading regions must be designed to reflect the shape of the underlying body. As shown in Fig. 3, we pre-define the loading region on the human model. For each loading region \( R_i \), we set the loading coefficient \( \tau = 1 \) at the center of \( R_i \) and \( \tau = 0 \) on its boundary. The loading coefficient inside \( R_i \) varies smoothly. Additionally, we set \( \tau = 0 \) in the non-loading regions.

The regions with \( \tau > 0 \) on the human model have to be in contact with the garment model. In Fig. 3, the regions with bigger loading coefficients are rendered in dark pink, while the regions with smaller loading coefficients are rendered in light pink.

Since the loading regions are designed to reflect the shape of the underlying body, a point \( v^G \) belonging to the loading region of the garment corresponds to a point \( p^H \) on the human body that is geometrically similar to \( v^G \). We define such point pairs as matching pairs, and they serve to further adjust the shape of the garment model.
5.2. Adjusting garment shape with matching pairs

We take the position and surface normal into consideration when searching for the matching point \( p_i^H \) for each vertex \( v_i^C \). Specifically, we are looking for a point \( p_i^H \) that is very close to \( v_i^C \) in terms of position and surface normal. To achieve this, we first find a set of candidate points \( \{ p_i^H \} \) in the vicinity of each \( v_i^C \), and then define a measurement between \( v_i^C \) and each point in \( \{ p_i^H \} \) as follows:

\[
\omega_{ij} = |p_i^H - v_i^C| e^{1-n_i^C n_i^C} \tag{6}
\]

Where \( n_i^C \) and \( n_i^H \) are the normal on \( v_i^C \) and \( p_i^H \). The candidate \( p_i^H \) that minimizes \( \omega_{ij} \) is the matching point for \( v_i^C \).

Local high-frequency geometry information, such as wrinkles, affects the surface normal severely, which might influence the quality of matching results. In order to reduce this influence, we compute the normal for each vertex on \( G \) from its smoothed copy \( G' \), which is obtained by smoothing \( G \) with a curvature flow-based method [53]. Mathematically, the smoothed copy of each \( v_i^C \) is computed as follows:

\[
v_i^C = \frac{1}{4\lambda} \sum_{(v_j, v_k) \in E} (\cot \alpha_j v_i^C - \cot \beta_j v_i^C)
\]

where \( h_i^C \) and \( n_i^C \) are the mean curvature and surface normal on \( v_i^C \), respectively; \( \lambda \) is a small positive number, and \( \alpha_i \) and \( \beta_i \) are the two angles opposing the edge \( v_i^C v_j v_k \) in the two triangles linking \( v_i^C \).

As shown in Fig. 4(a), a ruffled skirt is adopted to illustrate the influence of the surface normal adjustment on matching results. Fig. 4(b) shows the garment–human alignment result using skeleton driven volumetric deformation and Fig. 4(c) shows the smoothed garment copy.

To accelerate the matching process, we use an octree to search for the matching point \( p_i^H \) that is very close to \( v_i^C \). We take the position and surface normal into consideration when selecting the one that minimizes \( \omega_{ij} \) as the matching point for \( v_i^C \). The smaller the size of leaf nodes, the more efficient the point matching will be; however, leaf nodes that are very small may make the matching results to rely on the sampled result too much. If this happens, the matching point might be an undesirable local point. Accordingly, to avoid these undesirable local minima, we set the minimal size of the leaf nodes to 5 cm.

Eq. (6) sets up the correspondence between \( G \) and \( H \). However, some incorrect correspondence pairs may still exist. Generally, a correct correspondence pair \( v_i^C \) and \( p_i^H \) must have a similar length and direction as other correspondence pairs around \( v_i^C \). Therefore, we define the matching correctness coefficient \( c_i \) on \( v_i^C \) as follows:

\[
c_i = |p_i^H - v_i^C| - |p_i^H - v_j^C| - |v_j^C - v_k^C|, \quad n_i = \frac{d_i}{|d_i|}, \quad d_i = \frac{1}{n_i} \sum_{(p_j^H, p_k^H) \in E} |p_j^H - v_i^C|
\]

where \( n_i \) is the number of linking edges on \( v_i^C \). The larger the \( c_i \), the larger the difference between \( p_i^H - v_i^C \) and \( p_i^H - v_j^C \), \( p_j^H - v_i^C \), \( v_j^C - v_k^C \); the more likely the incorrect matching will occur between \( v_i^C \) and \( p_i^H \).

For illustration, we compute the average and the variance of \( c_i \), as shown in Fig. 4(e). In the figure, \( C_{\text{Aver}} \) and \( C_{\text{Vari}} \) are the average of \( c_i \), \( \tau \), and the variance of \( c_i \), \( \sigma_i \), respectively. In the \( C-S \) coordinate system, the \( S \) coordinate represents the size of vertices while the \( C \) coordinate represents the correctness coefficient, the latter of which is bound to a value of \( \tau + 3\sigma_i \). As shown in Fig. 4(e) and (f), the \( \tau \) and \( \sigma_i \) values computed from the adjusted garment surface normal are smaller than those computed from the unadjusted garment surface normal. Otherwise stated, computing \( p_i^H \) with the adjusted garment surface normal results in \( v_i^C \) whose local transformations are more likely to be similar to those of adjacent vertices than those resulting from computations performed using the unadjusted garment surface normal would be. We visualize the value of \( C \) on the garment surface with different colors, as shown in Fig. 4(e) and (f); the average value, the minimal value, and the maximal value are colored in green, blue, and red, respectively.

We choose the vertex set \( Z = \{ v_i^C | p_i^H \in \text{loading regions} \} \) to serve as the position constraints for Eq. (4). For each \( v_i^C \in Z \), we set its weight \( \lambda_i \) in Eq. (4) as follows:

\[
\lambda_i = \tau_{p_i^H} e^{1-c_i} \tag{8}
\]

where \( \tau_{p_i^H} \) is the loading coefficient on \( p_i^H \). Eq. (8) indicates that for \( v_i^C \in Z \), a smaller matching correctness coefficient and larger loading coefficient results in larger constraint weights in Eq. (4); this increases the likelihood of superposition between the vertex and its matching point \( p_i^H \) during the garment shape adjustment result.

Fig. 5 shows the resulting shape adjustment. As shown in the figure, the vertices on the garment model whose matching points are in loading regions are dragged onto the human model surface.
5.3. Resolving penetrations between G and H

Two types of penetration exist: in one, $v^G_i \in G$ is inside H, while in the other is $v^H_i \in H$ outside G. In the first case, assuming the vertex's matching $p^G_i$ is inside triangle $f^G_i$ whose surface normal is $n^G$, if $n^G \cdot (p^G_{ij} - v^G_i) > 0$, then $v^H_i$ is inside H. In the second case, we register G into the octree on H and compute $v^H_i$'s matching point $p^H_i$ on G. If $n^G \cdot (v^H_i - p^H_i) > 0$, $v^H_i$ is outside G. $n^H$ is the normal on $v^H_i$. Since most of the human model surfaces are smooth, their normal changes smoothly across the surface, so the simplified method for point inside–outside detection presented above is effective and efficient.

In the first case, where $v^H_i$ is inside H, its ideal matching position is $p^H_i$. Alternatively, for $v^G_i$ outside G with matching point $p^G_i$ on G, the ideal position of $p^G_i$ is $v^H_i$. $p^G_i$ is further represented via Eq. (1). If these position constraints are substituted into Eq. (4), then the penetrations would be removed. Figs. 7 and 10 show the penetration resolution result of Figs. 5(a) and (b), respectively.

5.4. Adjusting ease allowance between G and H

After the penetration resolution between G and H, some parts of G are tightly fitted on H. Such a “tight fitting” might not be appropriate for garment design, as ease allowance between garments and human bodies is usually required to make the garments comfortable. Ease allowance is usually controlled by the girth on breast, waist, hip, and so on; accordingly, we adjust the shape of G using Eq. (4) with position constraints obtained from garment–human ease allowance and garment–human contact.

Suppose $Q^1 = \{v^G_i\}$ is the set in which $v^G_i$'s matching point on H belongs to the loading region. For each $v^G_i \in Q^1$, its ideal final position superposes with $p^G_i$ so that G contacts H on $v^G_i$ in the final state. We control the ease allowance constraints by editing the shape of girth curves on G. For example, suppose the sampling point set on a girth curve is $Q^2 = \{p^G_i\}$: the expected position of $p^G_i \in Q^2$ can be computed from the target girth curve. We then substitute the position constraints in Eq. (4) with the position constraints obtained from $Q^1$ and $Q^2$, and assign them with different $\lambda_i$. Constraints from $Q^2$ reflect the intentions of the designer, and we treat them as comparative, rigid constraints. We typically set a large value to $\lambda_i |p^G_i - Q^2|$: in our system we set it to a constant value of 5.0. Additionally, while constraints from $Q^1$ mainly depend on the gravity of the garment, we treat them as soft constraints. Here, we set the value of $\lambda_i |v^G_i - Q^3|$ to be the loading coefficient on $v^G_i$'s matching point $p^G_i$.

In adjusting the ease allowance, the user is asked to insert a plane against the garment model; then the intersection is fit with a B-spline. This B-spline serves as the girth curve, and can be edited to obtain the target shape. In Fig. 6, we edit the shape of the garment girth curves on the breast and waist to make the dress became looser.

6. Experimental results and analysis

We tested the performance of our approach with several garment models and human models. For convenience, the human models shown in Figs. 7(b)–(d) and Fig. 9(a) are named as A, B, C, D, respectively. The other human models shown in Figs. 9 and 11 are all deformed from the human model D. The garment model shown in Fig. 7 is a ruffled skirt with rich local geometric details, whereas the evening dress shown in Figs. 8 and 9 has two layers on the chest part, which features intricate cloth surfaces. The garment models shown in Figs. 10, 13(a) and 14 are coat, sweater, and long coat, respectively. These human models and garment models serve to demonstrate that our approach can fit garment models onto human models in a large variety of shapes and poses.

These garment models were from different resources. The evening dress and the coat were initially designed for human model A based on 2D-to-3D scheme using a commercial software. The sweaters and the long coat were exported from Poser [54]. The ruffled skirt, on the other hand, was generated by stitching a vest and a short skirt into one piece, while these two clothes were initially designed for two different human models. That is to say, the ruffled skirt has no initial reference human model. These various garments, each being designed by different methods, serve to emphasize that our approach does not require the use of initial human reference models, making it more flexible for many applications than existing methods [11–13] in which initial reference human models are required.

Not only can our approach fit garment models with “standard” poses onto human models, it can also fit garment models with “non-standard” poses, as shown in Fig. 12. Our approach can fit multiple garment models onto one human model and form a multi-layered clothing, as shown in Fig. 13. In this process, the sweater and the coat are fit onto human model B separately, as shown in Fig. 13(b) and (c). Then the sweater model is treated as one part of the human model, and our penetration resolving method is used to eliminate the penetration between the coat and the sweater, resulting in Fig. 13(d).
For a garment model with multi-pieces, as shown in Fig. 14(a), the neighboring relationship between the adjacent pieces are preserved in the global alignment, since two individual points located in the same position have the same barycentric coordinates in tetrahedral mesh. Therefore, after the skeleton-driven volumetric deformation, the continuity between the adjacent pieces is unchanged, as shown in Fig. 14(b). However, the penetration resolving method is shell-based, and since there is no continuity implied for the adjacent pieces, seams between the adjacent pieces could occur after penetration resolution, as shown in Fig. 14(c). To avoid this problem, before garment fitting, we sew up the adjacent pieces to make the garment model to be one piece. Then our approach can effectively resolve the penetration between the garment model and the human model, as shown in Fig. 14(d).

With our geometry-based framework, it is possible to fit a garment model to human models of various shapes and poses. However, the final fitting result may still seem to be lack of realism due to the unnatural draping. A fact is that the loading regions undertake enough necessary deformation to match the corresponding human model regions, yet the fashion regions do not. The deformation of the fashion regions, which is supposed to exhibit most of the draping behaviors, relies more on physical factors (e.g. gravity) than on geometric ones. Since physical factors are missed from our geometry framework, realistic draping will not be easily forged for the new human model. To address this issue, we apply physically based cloth simulation as a post-processing step to improve the quality of fitting result. A dynamics model is built on triangular garment mesh according to [55]. The fitting result from the above framework is taken as a rest state of the garment, and it evolves subject to gravity for many steps of simulation, as shown in Fig. 15.

Our approach facilitates the customized garment design process, therefore will benefit the garment design industry and garment marketing. The whole process will work in the following manner: clients choose a template model of a certain style they favor, for example the one in Fig. 7(a); they then use our software tool to transform the template to get a fitted dress, as shown in Fig. 7(b)–(d). The 3D fitted dress model are cut into pieces and flattened into 2D patterns, as shown in Fig. 16(a), with the tools described in our previous research [56,57]. These 2D patterns will subsequently be used for fabricating the dress.

Fig. 7. Fitting a ruffled skirt onto individual human models: (a) ruffled skirt and (b–d): fitting the ruffled skirt onto human model A, B, C, respectively.

Fig. 8. Fitting an evening dress onto individual human models.
Additionally, clients also have the freedom to decide the properties of cloth, such as physical materials and textures, as shown in Fig. 16(b).

Our experiments ran on a computer with Intel Core i7 CPU 920 @ 2.67 GHz+4.0 GB RAM. The linear system of equations was solved by the CHOLMOD package [58]. Furthermore, in addition to

Fig. 9. Fitting the evening dress onto human model D in a variety of shapes and poses: (a–c) fitting the evening dress onto human model D in a variety of shapes and (d–f) fitting the evening dress onto human model D in a variety of poses.

Fig. 10. Fitting a coat onto individual human models.
Fig. 11. Fitting the coat onto human model D in a variety of shapes and poses: (a–c) Fitting the coat onto human model D in a variety of shapes and (d–f) fitting the coat onto human model D in a variety of poses.

Fig. 12. Fitting garment models with non-standard poses onto human models: (a) coat model from Fig. 11(f), (b) fitting the coat model onto human model B, (c) sweater model, and (d) fitting the sweater model onto human model B.

Fig. 13. Fitting multiple garments onto human model D: (a) sweater model, (b) fitting sweater model onto human model B, (c) fitting the coat model onto human model B, and (d) resolving the penetration between the sweater and the coat.
the process of manually adjusting the garment skeleton, which usually takes 1–2 min, the computational time for fitting garment models onto individual human models were also measured and listed in Table 1. From these experimental results, we conclude that the computational cost is more closely correlated to the number of triangles of the garment and the human models, but less relevant to the garment shape, complexity, or pose.

Though our method can handle complicated cases to some extent, there are still limitations to be tackled in the future:

(1) In our work, a tetrahedral mesh is used to skin the garment model, which is efficient and effective in most cases. However, in some cases, especially when two different garment parts are close in geometric distance, but far apart in geodesic distance (such as the left arm part and the body part shown in Fig. 17(a)), some tetrahedrons might intersect with both parts; in this case, our method might segment the tetrahedral mesh inappropriately, as shown in Fig. 17(b). Unnatural deformations could occur in these regions, as shown in Fig. 17(c) and (d). Such unnatural deformations can be avoided if the tetrahedrons are small enough. For example, smaller tetrahedrons as shown in Fig. 17(e) result in more appropriate deformation as shown in Fig. 17(f) and (g). However, smaller sized tetrahedrons usually mean an increment in the number of tetrahedrons needed. Accordingly, the computation cost increases.

An alternative way of decreasing the dimension of the tetrahedrons is using the silhouette of the garment model to cut the tetrahedrons intersecting with different garment parts into separately sub-parts, thus we can use fewer tetrahedrons to correctly tetrahedralize the garment model. We will verify this method in the future.

(2) Our penetration resolving method is not powerful enough for some complex situations. When the penetration occurs on intricate human surfaces, such as the crossed hands in Fig. 18(a), it might fail to resolve the penetration between the human models and the garment models. Another extreme case is that the pose of the garment model differs too much from that of the human model, leading to a large deformation for the garment model in the initial alignments and the penetration situations become very complex. Fig. 18(b) shows such an example in which the evening dress model is to be fit onto human model D in a pose of ready-for-run. Our current method is hard to resolve such penetrations.

(3) We have only considered the penetrations between garment models and human models. Self-penetrations inside the garment models, such as the case shown in Fig. 18(a) where the arm part intersects with the body part, have not been tackled yet.

(4) We control the garment–human ease allowance by editing the shape of girth curves. More flexible interactive operations, such as sketch-based mesh editing [59] will be considered in the future.

7. Conclusion and future work

In this paper, we proposed a novel framework for fitting 3D garment models onto various human models. The framework is based on a variety of techniques, such as skeleton-driven volumetric Laplacian deformation, garment–human surface matching and evaluation, garment–human penetration resolution, and garment–human ease allowance adjustment. Our skeleton-driven volumetric deformation makes it possible to fit a variety of garment
Fig. 16. Integrating with our previous techniques [56,57] for application in garment design: (a) designing the garment pieces directly on the 3D garment model, then flattening the 3D pieces into 2D pieces (represented by mesh patterns). (b) Using texture mapping to visualize the properties of cloth.

Table 1
Experimental results.

<table>
<thead>
<tr>
<th>Fig.</th>
<th># Faces in G</th>
<th># Vertices in G</th>
<th># Faces in H</th>
<th># Vertices in H</th>
<th>Computing time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 7</td>
<td>8445</td>
<td>4401</td>
<td>19,994(6d)</td>
<td>9999</td>
<td>17.7 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13,598(6b)</td>
<td>6799</td>
<td>16.7 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>39,961(6c)</td>
<td>20,000</td>
<td>27.9 s</td>
</tr>
<tr>
<td>Fig. 8</td>
<td>5762</td>
<td>3009</td>
<td>19,994(7d)</td>
<td>9999</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13,598(7b)</td>
<td>6799</td>
<td>15.8 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>39,961(7c)</td>
<td>20,000</td>
<td>24.3 s</td>
</tr>
<tr>
<td>Fig. 9</td>
<td>5762</td>
<td>3009</td>
<td>52,978</td>
<td>26,521</td>
<td>40.1(9a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.6(9b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42.5(9c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.3(9d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.8(9e)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>42.1(9f)</td>
</tr>
<tr>
<td>Fig. 10</td>
<td>3953</td>
<td>2060</td>
<td>19,994(6d)</td>
<td>9999</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13,598(6b)</td>
<td>6799</td>
<td>16.6 s</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>39,961(6c)</td>
<td>20,000</td>
<td>23.2 s</td>
</tr>
<tr>
<td>Fig. 11</td>
<td>3953</td>
<td>2060</td>
<td>52,978</td>
<td>26,521</td>
<td>36.2(11a)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37.1(11b)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>39.6(11c)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>35.8(11d)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>37.5(11e)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>40.2(11f)</td>
</tr>
<tr>
<td>Fig. 12</td>
<td>3953(11a)</td>
<td>2060(11a)</td>
<td>13,598</td>
<td>6799</td>
<td>36.2(11a)</td>
</tr>
<tr>
<td></td>
<td>4321(11c)</td>
<td>2241(11c)</td>
<td></td>
<td></td>
<td>37.1(11b)</td>
</tr>
</tbody>
</table>

Fig. 17. The size of the tetrahedrons influences the fitting result: (a) the coat model where the left arm part is closer to the body part than the right arm, (b) tetrahedralizing the coat with relatively bigger sized tetrahedrons; some tetrahedrons intersect both the body part and the arm part of the coat. (c, d) Fitting the coat onto a human model, large deformation occurs near the armpit. (e–g) Decreasing the size of the tetrahedrons can reduce the unnatural deformation.
models onto a broad spectrum of human models in a variety of shapes and poses. Furthermore, our Laplacian graph deformation guarantees that the local details of the garment surface will be kept well within the fitting results. Using the constraints of garment–human contact and those of garment–human ease allowance, our framework converts garment design knowledge into geometric constraints, rendering the fitting results more realistic. The technique developed in this paper additionally improves the reusability of garment models, and is therefore very useful for the garment design industry.

Acknowledgements

We appreciate the anonymous reviewers for their helpful comments and suggestions. This work was supported by the National Natural Science Foundation of China (NSFC, No. 60903145), Beijing Natural Science Foundation (4102062), National High-tech R&D Program (863 Program) (2009AA04Z103), and the Open Project Program of the Engineering Research Center of Digitized Textile & Fashion Technology, Ministry of Education, Donghua University. We would like to thank Digital Fashion Ltd. for providing the evening dress model, the coat model, and the human model A.

References


Fig. 18. Failing to resolve penetrations: (a) the penetration between the intricate human surface and garment model, and even as the self-penetrations inside garment models, are hard to be resolved. (b) When the pose of the human model is extremely different from that of the garment model, penetrations between the garment model and the human model are hard to be resolved.


Poser. (http://poser.smithmicro.com/).


