Multimed Tools Appl (2012) 59:973–990
DOI 10.1007/s11042-011-0781-6

On generating realistic avatars: dress in your own style

Hui Yu · Shengfeng Qin · Guangmin Sun · David K. Wright

Published online: 16 March 2011

Abstract The use of 3D avatars is becoming more frequent with the development of computer technology and the internet. To meet users' requirements, some software or programs have allowed users to customize the avatar. However, users are only able to customize the avatar using the pre-defined accessories such as hair, clothing and so on. That is, users have limited chance to customize the avatar according to their own styles. It will be of interest to users if they are able to change the appearance of the avatar by their own design, such as creating garments for avatars themselves. This paper provides an easy solution to dressing realistic 3D avatars for non-professional users based on a sketch interface. After a user drawing a 2D garment profile around the avatar, the prototype system can generate an elaborate 3D geometric garment surface dressed on the avatar. The construction of the garment surface is constrained by key body features. And the garment shape is then optimized to remove artefacts. The proposed method can generate a uniform mesh for processing such as mesh refinement, 3D decoration and so on.

Keyword 3D avatar · Avatar customization · Virtual dressing · Virtual world

1 Introduction

The word "Avatar", originating from a Hindu Sanskrit term [17], means "A deity in visible earthly form". With the development of information technology, "Avatar" has been used to

H. Yu (✉) · S. Qin · D. K. Wright
School of Engineering and Design, Brunel University, Uxbridge UB8 3PH, UK
e-mail: hui.yu@brunel.ac.uk

S. Qin
e-mail: sheng.feng.qin@brunel.ac.uk

D. K. Wright
e-mail: david.wright@brunel.ac.uk

G. Sun
School of Electronic Information and Control Engineering, Beijing University of Technology, 830022 Beijing, China
e-mail: gmsun@bjut.edu.cn

refer to the representation of a human being in a virtual environment [7, 8, 10, 11, 27]. In recent years, avatars have been used for different purposes, such as social virtual worlds (Second Life [28], CyberTown [7] and [9] etc.), video games and entertainment programs (Kinect [16], Grand Theft Auto: San Andreas [11], Half-Life [12] and Wii [38]), instant-messenger (such as Windows Live Messenger, Google Talk, Yahoo instant messenger etc.), virtual meetings and conferences, and so on. In social virtual worlds, avatars representing users in the virtual world interact with each other, such as participating in events, socializing etc. In video games, avatars physically represent players in the virtual game world. In instant-messenger, an avatar is used to represent the user while chatting online with other users. Especially with the development of the internet, the use of avatars is growing more prevalent. It is inevitable that users will require customizable avatars, which allow them to change appearance such as hair style, body shape, dresses etc. Some virtual world software allows users to customize avatars, but users can only choose predefined products such as hair. Can users customize avatars using their own design, for instance, dressing avatars with their own styles? This paper presents a method for customizing avatars' clothing. The target application of the proposed method focuses on social virtual worlds, virtual conferencing, and instant messenger.

The objective of this paper is to enable users to interactively dress 3D avatars with their own styles instead of choosing clothing from a database. Figure 1 shows an example of generating a 3D sweater using the proposed method and the prototype interface. To improve realism, the avatar body was scanned from a real human being. In this paper, realism means two aspects: one is the 3D avatar body, which is from scanned data of a real human body; the other is about the generated garment. The avatar head can be improved using some existed methods [3, 34]. One possibility is that users could use their own body scans both to create avatars for popular programs such as Wii Fit Plus [38] and to generate a set of virtual garments for those avatars. This paper focuses on providing an easy-to-use method for users to create convincing virtual garments for 3D avatars given scanned body data. For specific applications, further work needs to be done to adapt the method to the target application. To meet the increasing needs for accessing multimedia content for non-professional users, some easy-to-use interfaces have been developed [13]. The proposed approach adopts the sketch-based interface. A sketch-based interface is one of the most convenient ways for users to convey 3D shapes using computers. Many sketch-based

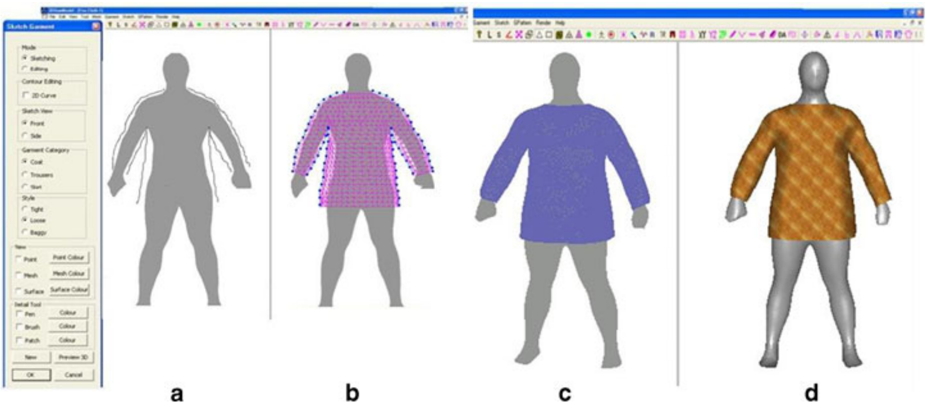


Fig. 1 An example of a 3D sweater generated by our sketch-based system. **a** Sketch the profile of a sweater. **b** Complete mesh construction. **c** Mesh surface after refinement. **d** Rendering result

systems have been developed for various applications such as the Teddy system [15] for easily creating 3D free-form models and the interactive sketching system [4] allowing users to directly sketch in 3D. The proposed approach takes user-drawn strokes as input in 2D space and generates a geometric garment dressed on the avatar's body in 3D space. Key body features and various optimization schemes are applied to help obtain a pleasing garment surface.

Important features of the system are as follows:

Key body features: key body features are used as constraints during garment generation. Key body features are extracted through finding extreme points or girth of body cross sections.

Sketch interface: users are allowed to draw strokes on 2D screen, which represent the front view shape of a garment. These strokes are then processed into spline curves (see Fig. 1). Through dragging control points into new positions, the garment shape can be modified accordingly.

Shape optimization: a level set-based method is used to optimize garment shapes.

Garment surface refinement: This is obtained by applying some mesh refinement schemes, which improves the garment shape and allows some further mesh editing tools such as decoration tools to be applied.

Decoration Tools: a 3D embroidery tool allowing users to add details to the garment by sketching on the garment surface. Decoration details are represented by a refined mesh surface.

2 Related work

Several methods have been proposed for dressing virtual humans borrowing the traditional garment-making method in the real world: creating virtual garment patches and sewing up. Interactive techniques to place clothing patterns on virtual actors have been presented to simulate dressing [35] and Arnulph et al. [1] described an interaction-free method for pre-positioning cloth patterns on a 3D scanned human body. The limitation of the above mentioned methods is that users are only allowed to choose very limited pre-defined virtual garment patterns or patches, which prevents users from dressing virtual humans with their own designs.

Wang et al. [37] developed a sketch-based method for garment design, which uses a feature template for generating 3D garments. However, it only allows users to locally specify garments profile through 2D sketches, giving users limited flexibility to sketch the overall garment profile, and it only uses body features as reference for sketching horizontal cross-sectional profiles of a garment (see Fig. 2). The method proposed in this paper allow users to sketch vertical strokes as garment profiles and use human body features as constraints during the generation of vertices for 3D garments (see Fig. 4). Turquin etc. [32, 33] proposed an approach for dressing virtual characters with complete geometric garments based on sketches. Through drawing garment profiles around a 3D human model, the system can generate various kinds of garments based on the distances from user-drawn strokes to the human model. To remove artifacts from garments, it uses Bezier curves in horizontal planes to interpolate the z-values for the garment part between two limbs to mimic cloth tension. The authors explored a method for geometrically dressing virtual humans, but did not provide any algorithms for processing garment shape around torso parts such as the back, chest and stomach. As a result the output garment follows the body

Fig. 2 Sketching garment profile proposed by Wang et al. [37]

shape too closely over these parts. In addition, its output is a simple garment mesh which requires further processing. Igarashi etc. proposed an intuitive method for clothing manipulation [14]. Users first edit 2D cloth patterns and then put cloth on the 3D body through establishing proportional correspondence of free-form mark pairs drawn by the user on both the target body and the clothes. Once a garment has been put on the body, users can then manipulate garment movement with surface dragging. In their method, cloth patterns are created separately before being dressed on the 3D body.

In this paper, we propose a method for dressing 3D avatar automatically once garments are generated through user sketching input. The proposed method uses a level set-based method to modify the curve shape of garments, which prevents garments from exactly following the shape of the body. A garment surface is refined using mesh smoothing and subdivision schemes, which improves gives flexibility for further processing such as providing 3D decoration tools.

3 Dressing method

3.1 Avatar body pre-processing

The raw data from the body scanner is range data without semantic information needed for avatar dressing. It needs to go through a pre-processing procedure before the dressing algorithm is applied. The simple pre-processing procedure extracts eight key body features, which are used as constraints for garment generation and dressing. Key body features are obtained through calculating the extreme points or circumference on the avatar body after body segmentation [39] (Fig. 3(a) human model with boundaries (b) front hanging points (c) back hanging points). To make it easy for users to sketch a garment profile for the avatar, we need for the avatar's arms to be a clear distance from the main body and for a

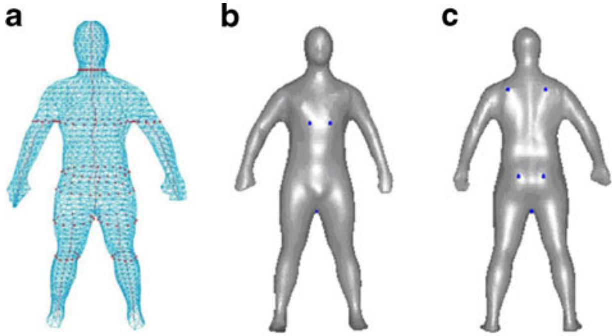


Fig. 3 Key body features. **a** human model with boundaries **b** front hanging points **c** back hanging points

clear distance between the two feet as demonstrated in Fig. 3. Table 1 classifies those key body features according to constraint types as follows:

Here is an explanation of constraint types, which also briefly indicates how those key features work as constraints.

> *Boundary:* marks the boundary for 3D construction. If a stroke (outer contour curve) is beyond the boundary, the segment of the stroke beyond the boundary references the boundary for 3D construction.
> *Hanging Point:* represents the maximum or minimum value along the z-axis of a body segment where a garment touches the human body. Front and back hanging points mean the point on the front side and backside of the avatar body.
> *Interior boundary:* marks an interface for dramatic change of the body shape. Some pre-defined thresholds such as curvature and curve energy need to change accordingly when using the level set-based method.

3.2 Stroke processing

Raw sketch strokes drawn by users are processed into 2D curves and associated with different segments after classification. There are two stages for stroke processing: *spline curve fitting* and *curve classification*.

> *Spline Curve Fitting:* The sketch strokes are first converted into 2D curves using spline fitting [26] (Fig. 1(a)–(b)). There are several advantages brought by the curve fitting. First, noise in the raw strokes has been filtered. Second, the large number of raw stroke points has been significantly reduced and well distributed with evenly spaced intervals, which speeds up 3D point construction. Third, users are allowed to draw as many strokes as they like corresponding to one body segment. These strokes do not need to be drawn in sequence, and it allows the user to leave a gap between two strokes. The system can group those strokes by averaging x-coordinates of those points within the small scope of each sampled y value along y-axis and fit these points from averaging with one spline curve. Finally, due to the property of the spline curve, it can be easily modified by users to obtain different shapes of garments.
> *Curve Classification:* After curve fitting, these curves are classified into two categories: inner contour curve, denoted by *s_nc,* and outer contour curve, denoted by *s_oc* (Fig. 4(a)). Each curve is associated with a body segment after recognition. If a curve intersects with the human body and the intersection points exceed a certain number, it is classified as an inner contour curve. Otherwise it is an outer contour

Table 1 Key body features

Key feature	Constraint type	Denotation
Neck	boundary	b_neck
Nipple	Front hanging points	h_nipplel, h_nippler
Scapula	Back hanging points	h_scapulal,h_scapular
Waist	Interior boundary	b_waist
Wrist	boundary	b_wristl, b_wristr
Hip	Back hanging points	h_hipl, h_hipr
Crotch	Boundary points	b_crotch
Ankle	Boundary	b_anklel, b_ankler

curve. In Fig. 4(a), the curves constituting a skirt strap are inner contour curves and those which are away from the body are outer contour curves.

Once curve classification have been achieved, garment surface construction can start using the techniques described in the following sections.

3.3 Garment surface construction

Two types of constraints are used during the generation of a 3D garment position. One is from human body features, the other is the distance between the projected 3D curve and the body surface. The distance d_o_i between an outer contour curve s_oc and its corresponding body surface is used as a reference to determine the interior garment vertices. The system converts garment profile curves into a proper 3D garment surface using the following five steps.

Step 1 Generation of garment vertices: Rough 3D vertex positions are determined based on garment contour curves. Methods are developed for inner contour curves and outer contour curves separately (Fig. 4(c) and (d)).
Step 2 Garment shape optimization: Rough 3D vertex positions are optimized to improve the realism of a garment (Fig. 1(d)).
Step 3 Garment mesh generation: A mesh algorithm is applied directly to 3D garment vertices to construct a coarse mesh of garment segments which are sewn up to obtain a complete garment surface.
Step 4 Garment mesh refinement: An elaborate mesh is obtained using smoothing and subdivision schemes.
Step 5 Shape modification: Through dragging control points of spline curves, which represent the front shape of a garment, garment shape can be modified without efforts to re-sketch a new one (Fig. 9).

3.3.1 Generation of garment vertices

The generation of a garment's vertices is based on the distances between the body and garment contour curves. There are two types of contour curves (Fig. 4(a)) as follows:

- *Inner contour curve*
- *Outer contour curves.*

Different algorithms are developed to generate 3D garment vertices for these two types of curve. Each curve has its region of interest (ROI) formed by the curve and the horizontal line going through the end point of the curve. If two kinds of ROIs overlap, only the ROI from the outer curve is valid in the overlapping region. There are two types of garment vertices generated in each ROI which are silhouette vertex and interior vertex. The inner contour curve is used for generating interior vertices within its ROI (p_nj and v_nfi in Fig. 4 (b)) and the outer is used for generating both silhouette vertices (p_oj in Fig. 4(b)) and corresponding interior vertices (v_oi in Fig. 4(b)) . Vertex generation from both inner contour curves and outer curves is described in detail as follows.

- Garment Vertices from Outer Contour Curves

There are two steps for vertex generation from the outer contour curve. This method is applied to the curves corresponding to arms, chest, torso and legs.

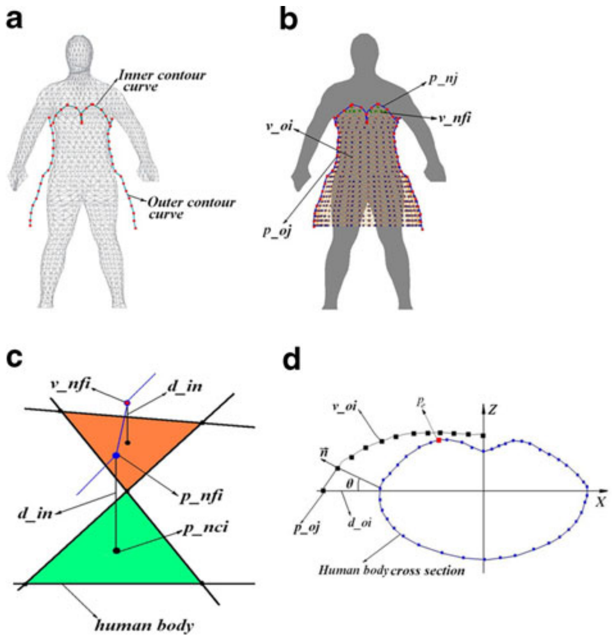


Fig. 4 Garment vertex generation and contour curves where *Pe* is a hanging point as a constraint

Step 1 Generate silhouette vertices.

We calculate the distance d_o_i from the outer contour curve, denoted by $s\_oc = \{p\_o_1, p\_o_2, ...p\_o_j...\}$, to the human body (Fig. 4(d)). The point depth of an outer contour curve is set to the z-value of the nearest point of the human body.

Step 2 Generate interior vertices.

To infer the rough position of interior vertices v_oi of a garment (Fig. 4(b)) based on the distance d_o_i obtained in step 1, we use the following equation to specify the distance between the body curve and the interior garment's vertices (Fig. 4(d)).

$$dr_i = d\_o_i{}^* \cos(\theta)^* \lambda + \delta \tag{1}$$

Where θ is the angle between the normal of a body curve at point p_i and the x-axis ($\theta \leq \pi/2$), δ is a small offset variable defined by the user ($\delta \geq 0$), and λ is a coefficient used to adjust the distance through experiments (usually $\lambda \leq 1$). When the z coordinate of a moving point is equal to that of point *Pe* (Fig. 4(d)), which is a feature point on the cross section, the value of δ should be increased to make a more flat curve rather than exactly following the body shape.

- Garment Vertices from Inner Contour Curves

Generation of garment vertices from an inner contour curve is straightforward compared with those from an outer contour curve.

Step 1 Generate silhouette vertices.

Through the use of ray tracing techniques, a set of points denoted by $s\_nc = \{p\_nc_1, p\_nc_2, ...p\_nc_i...\}$ are obtained by the intersection between an inner contour curve and the human body. And then *s_nc* is offset to *s_ncf* by a distance *d_in* from the human body along each triangular face that each point of *s_nc* is projected onto (Fig. 4(c)). The offset inner contour curve *s_ncf* is represented by $s\_ncf = \{p\_ncf_1, p\_ncf_2, ...p\_ncf_i...\}$. The distance *d_in* is usually a pre-defined small value, and it is subject to some constraints. For example, *d_in* takes zero if the projected points offset from nipples using constraints *h_nipplel* and *h_nippler* or other hanging points, which means that the garment touches nipples or hanging points.

Step 2 Generate interior vertices.

Interior garment vertices, v_nf_i, within the boundaries are interpolated using the distance field of the human body (green points in Fig. 4(b)). The same distance parameter, *d_in,* is used during the interpolation (Fig. 4(c)). Constraints are applied to ensure that interior vertices do not lie outside of hanging points, which uses the same principle as in step 1.

Figure 5 shows an example of a mesh generated using the above method.

3.3.2 Garment shape optimization

Through interpolating 3D points based on distance to a human model and constraints, we have obtained a rough shape of a garment. Since distances from the profile curves to the body were used to refer positions of the vertices of a garment, the surface of the body might affect the shape of a garment, especially the area along the spine on the back of a human body. Thus, in this subsection, a technique based on the level set method is proposed to optimize the shape of a garment, such as removing artefacts and mimicking cloth tension The points in the spine area of a garment are selected as points of interest by calculating their signed curvature.

To help understand the shape processing, a brief introduction to the level set method is given here. The level set method, devised by Osher et al. [22], is a versatile method for computing the motion of a curve. The purpose of this method is to calculate the subsequent motion of the curve using a velocity field. An oriented curve evolving in time is represented by the zero level set of a level set function. Curve $C(t) : \Re^n \mapsto \Re$ represents implicitly the zero set of the function $\phi(C^*, t) = \pm d$.

Where $C^* \in \Re^n$ and the variable d is a signed distance from C^* at time t to C(t). The orientation is denoted by defining the areas in which the level set function is negative in interior and positive in exterior.

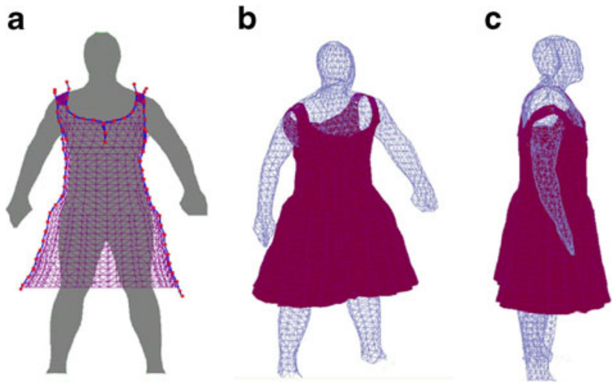


Fig. 5 Garment generation. **a** Mesh model. **b** Front view. **c** Side view

The level set function, Φ, is defined in a domain Ω ($\Omega \in \Re^n$). The initial curve *C(t)* is evolved based on a speed field F. This is usually represented by the equation:

$$\frac{\partial C}{\partial t} = F\vec{N} \tag{2}$$

Where F denotes the velocity field and $\vec{N}$ is the normal of the curve. This velocity can depend on two kinds of properties: internal and external properties. Internal properties include curvature, position, geometry etc. External properties include time, external physics and so on. The curve is obtained later as the zero level set of a smooth function. If the level curves propagate at a constant velocity, the level set $\{\phi = d | d \in \Re\}$ is a set which is away from the original curve with a distance *d*. In this occasion, the motion of level curves is subject to Huygens–Fresnel principle (Fig. 6(a)). From Eq. 2 we can see that to evolve the curve we need to solve two problems: one is the evolution direction of the curve at each point, the other is the moving speed of the point.

In this paper, the shape of a curve is optimized by moving points of interest, which are identified by calculating their curvatures. Considering the discrete representation of the zero level set, points of the curve are moved along its normal direction. The evolution of a curve is implemented through the speed field F. A positive velocity means that the point moves outwards, while a negative velocity means that the point moves inwards. The signed curvature of point p_i is obtained using Eqs. 3 and 4. The next position of the moving point is determined by the relative distance along its normal direction. The shape optimization process is achieved using following four steps.

Step 1 Calculate the signed curvature k^* at each point.

Since the garment curve is represented by discrete sets of points, the curvature at point p_j can be obtained from the following equation [21]:

$$k(p_i) = \frac{2\sin(\frac{\beta}{2})}{\sqrt{\|v_1\| \bullet \|v_2\|}} \tag{3}$$

Where, $v_1 = p_i - p_{i-1}, v_2 = p_{i+1} - p_i$ and p_i, p_{i-1}, p_{i+1} are three consecutive points on a curve. And β is the angle between the vector v_1 and v_2.

The signed curvature k^* is defined as follows:

$$k^* = \begin{cases} k(p_i), & \text{if point pi is a convex point;} \\ -k(p_i), & \text{if point pi is a concave point.} \end{cases} \tag{4}$$

Curve energy: can be obtained by the equation:

$$E = \sum k^2 \tag{5}$$

Where k is the curvature of a curve at a point.

Step 2 Identify points of interest on curves of a garment.

Points of interest are identified using curvature at each point. For example, if the curvature of a point is larger than the pre-defined value then it is taken as a point of interest.

Step 3 Calculate the normal of each point.

Garment vertices obtained from methods described in the previous section constitute a series of cross sections, which can be seen as closed or open curves of 2D domain, represented by C_i. The normal of the cross section curve at a point p_j can be obtained by calculating the gradients of the cross section curve at this point.

Step 4 Calculate the velocity field of the a point using the equation: $F = v \cdot \vec{n}$. The variable v is the speed of each point of interest for movement.

To simplify the propagation of the discrete curve *C(t)* we define the speed as follows:

$$v = \begin{cases} 1, & \text{if } p_i \text{ is a point of interest;} \\ 0, & \text{if } p_i \text{ is not a point of interest.} \end{cases} \tag{6}$$

Where k is the curvature at point p_i and w is a coefficient.

Two criteria are used during the optimization: signed curvature threshold, curve energy threshold and body features. The values of curvature and energy threshold are determined by experiment. If the sign curvature at point p_i is greater than the threshold ($k^* \geq threshold\_curva$) then p_i strops moving. If the curve energy exceeds the pre-defined value ($e \geq threshold\_eng$) the curve stops evolving. Figure 6(a) shows curve evolving constrained by curve energy and curvature and (b), (c) and (d) demonstrate various tensions in lower parts of a skirt. Curve energy below the crotch in (c) and (d) are 50 and 10% of (b) separately. Body features together with curvature thresholds are used to constrain a garment shape. Points of interest of a curve stop moving when either feature constraints or curvature threshold are met.

3.3.3 Mesh generation

The 3D garment vertices are generated corresponding to each part of a human body. Thus, these vertices have already been separated into several parts. Two steps are involved in the mesh construction algorithm. In the first step, a correspondence triangulation algorithm is applied to each part. And then the surface of all those parts or patches of a garment are sewn together.

Step 1 Mesh generation for each part.

The shape of each part of a garment is relatively simple compared to the whole garment. Most of them are of a tubular or strap-like shape. A strap-like shape can be seen as a stack

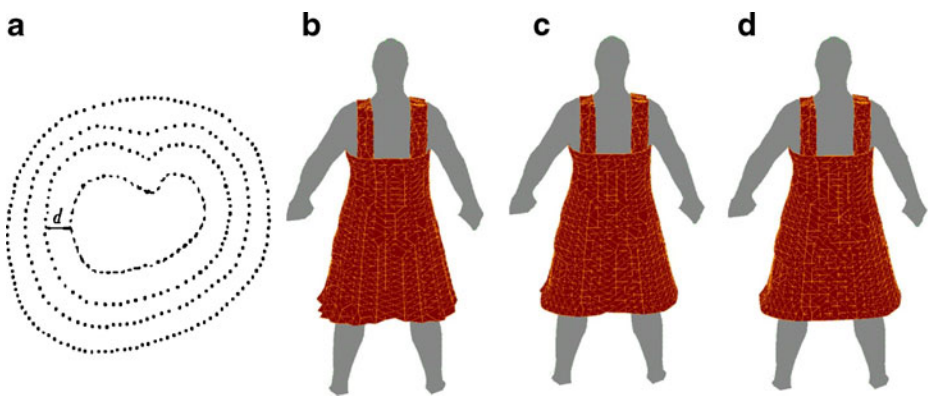


Fig. 6 Garment tension mimic

of open contours. To create mesh for the tubular and strap-like shape part, we use the correspondence triangulation method [20].

Step 2 Surface part connection.

After completing triangulation of all parts, an algorithm is applied to sew them together to create a complete garment surface. Two kinds of connection approach are used in terms of the shapes of the parts to be sewn: tubular-strap shape sewing and tubular-tubular shape sewing. For the first situation, there are usually more than two contours in the neighbouring planes (Fig. 1(b)). Normally, this happens to the connection among the chest, arms and the torso, which consisting of four contours. We use the arbitrary topology shape construction method [2].

The second situation happens to strap skirts or cocktail skirts (Fig. 8). After constructing the lower parts of a skirt and the strap or top upper parts of a skirt, we just need to sew the bottom edges of the strap or the bottom edge of the upper part of a cocktail skirt and the top edge of the torso surface. Two boundary points should be interpolated on the top edge of the torso surface. The point interpolation is straightforward. The boundary curve is projected onto the torso surface along its normal direction. The intersection between the projected curve and the edge of the torso surface is the interpolated boundary node. Figure 7 demonstrates the sewing algorithm with and without the boundary node interpolation, where the hollow points in (b) are interpolated boundary nodes. Then the adjacent points are connected by using the correspondence triangulation method [20].

Generally, a complete skirt mesh surface is achieved by interpolating boundary nodes and sewing up mesh patches above and below the armpits. Figure 8(a) shows an example of sewing skirt straps with the lower part without boundary node interpolation at the top edge below the armpit. After interpolate nodes along the normal direction of the nearest triangular face on the lower part, we can obtain a complete garment mesh (Fig. 8(b) and (c)).

3.3.4 Garment mesh refinement

After constructing a complete garment surface, the visualisation of the constructed garment surface may not be acceptable. To improve visualization and get an elaborate garment mesh for further processing or simulation, we apply some mesh refinement schemes.

We smooth the coarse garment surface using an improved Laplacian algorithm [36]. And then the smoothed mesh is refined through applying a subdivision algorithm. We use a generalization of the subdivision scheme introduced by Loop [19], which is one of the

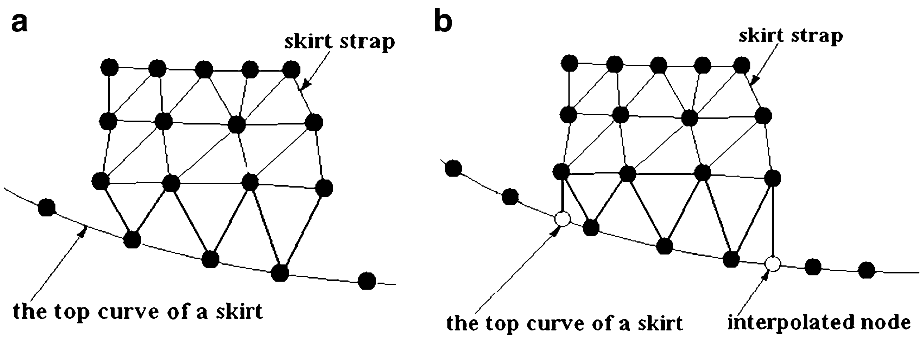


Fig. 7 Node interpolation. **a** Sewing before interpolating boundary nodes. **b** Sewing with interpolating new boundary nodes

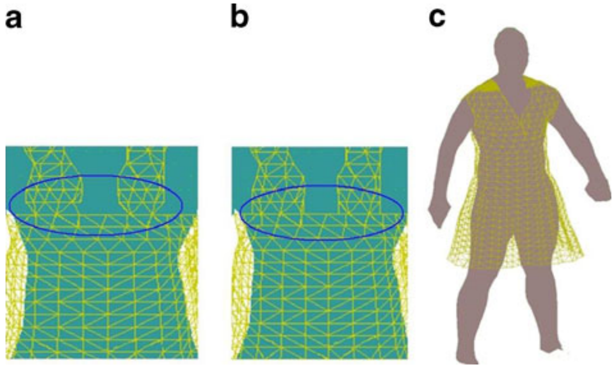


Fig. 8 Examples of sewn mesh of skirts

simplest algorithms generating tangent plane smooth surfaces. To avoid boundary shrinkage we apply a boundary mask for boundary vertices of a garment mesh during subdivision [40]. An example of garment mesh subdivision is shown in Fig. 9.

3.3.5 Shape modification

The characteristics of spline curves allow the profile shape of garments to be easily modified (Fig. 10). Users can modify the profile shape of a garment by dragging the control points of spline curves to the desired position in both 2D and 3D spaces. Through moving the control points of existing curves, users are not disturbed by accidently changing the whole shape. This allows them to obtain various shapes for comparison through modification.

3.4 Penetration detection

Penetration between the 3D human model and garment surfaces might happen during the procedure of garment mesh construction, shape optimization and mesh processing, For example, penetration occurs between a T-shirt mesh refined from the original skirt mesh and the human model (Fig. 11(a)).

We developed a layer-distance-based algorithm to perform penetration detection and response. Both human models and garments are represented by surfaces, which are marked as different layers. We check penetration by calculating distances between the two layers.

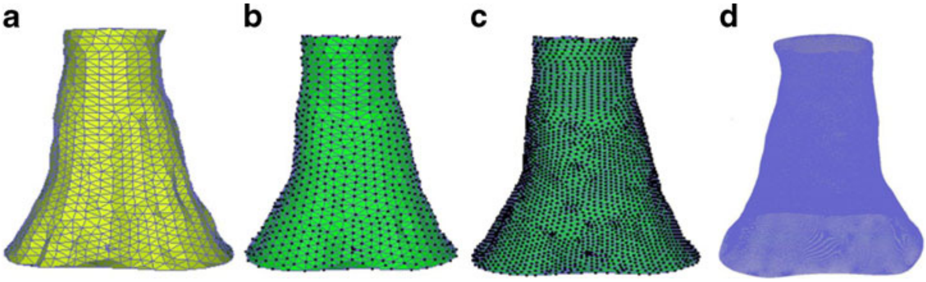


Fig. 9 Mesh smoothing and subdivision. **a** The original mesh. **b** Mesh smoothing using the improved Laplacian method. **c** Loop subdivision:1 iteration. **d** Loop subdivision:3 iteration

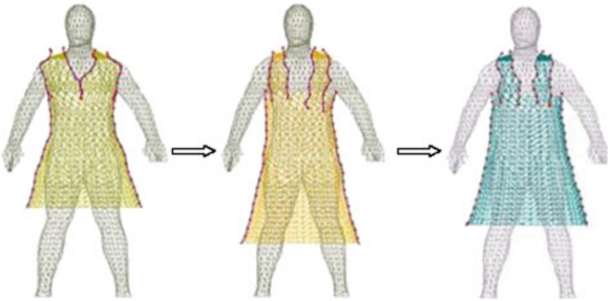


Fig. 10 Examples of garment shape modification

The inner layer is usually taken as the reference layer during the procedure. We compute signed distances denoted by d_c from vertices of a body to the garment surface represented by S_g along the normal direction of body vertices. If $d_c \geq 0$, there is no penetration, otherwise there is penetration.

If penetration occurs, related vertices of the garment are moved outwards by Δd in the direction of the face normal. The value of Δd is determined by the equation $\Delta d = |d_c| + \sigma$. The variable σ is a small tolerance.

An example of penetration response is shown in Fig. 11(a)–(c).

3.5 Decoration tools

This system provides 3D decoration tools to allow users to add details to garments. In reality, almost all garments have details such as a brand name, brand label and some colourful strap etc. In our current system,

The embroidery decoration consists of two surfaces: the top surface and the side surface. The side surface is constructed by the two boundary curves P and Q. The top surface is based on the raised boundary curve Q and some interior points of the curve Q which is generated by the Lawson's incremental insertion method [18].

The raised boundary curve Q is projected onto a proper plane in its major direction using the PCA method. In this case, the 3D boundary curve Q is converted into the 2D domain denoted by Q_p. Through applying 2D Delaunay triangulation to the projected curve Q_p, which is a planar straight line graph, we obtain a rough triangular mesh. Then constrained Delaunay triangulation and Ruppert's refinement algorithm are used to generate quality

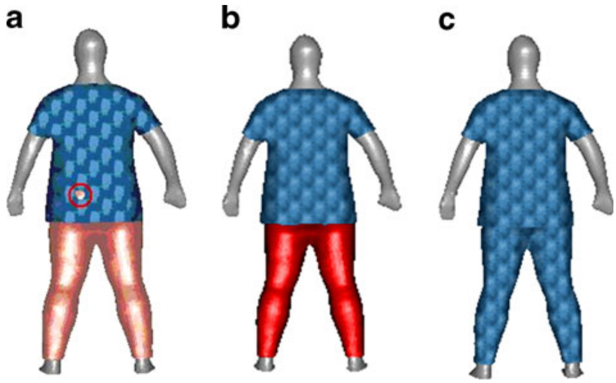


Fig. 11 Examples of penetration detection

surface [24, 29, 30]. A toy pinwheel pattern, and its symbol and some arbitrary sketching patterns are embedded on a skirt surface using this embroidery tool shown in (Fig. 12).

4 Implementation and user feedback

A prototype system has been developed using C++, OpenGL graphics API and GPU programming to illustrate the idea. A user study was designed to test the system, which involved ten participants (five males and five females). All participants were familiar with or had experience of using computers, but none of whom had experience of using sketch interface. Among those participants, there were five design graduates, one engineering designer, two social science graduates, one graphics designers and one animator.

Each participant was given 3 min of verbal tutorial before starting to perform two tasks as follows:

Task 1 participants were asked to create two garments using the prototype system according to images of five different garments shown in front view. And then they were required to change each of those generated garments into three different shapes with their own design.

Task 2 participants were required to create three garments completely by their own design.

During the procedure of both tasks, participants were allowed to modify the garment shape before finishing each one. All participants finished two tasks in 30 min. For task one, it took each participant about 3 min on average to generate a virtual garment according to the given image. For task two, participants used much time to modify the garment shape to meet their desirable design. Among those participants, seven people used pen frequently for sketching in work and the rest were usual users. During the testing, we found that six of those seven participants with pen-sketching experience had noticeable advantages over the others in terms of average finishing time. The graphic designer had the shortest finishing time. This may reflect the similarity between using the proposed method and the designer's conventional skills for designing or creating dresses for avatars. Though participants who did not have much pen-sketching experience took longer to finish the task than those with pen-sketching experience, they still found the system was easy to use.

Comparing the two tasks, we can see that once the garment shapes are given, users can easily create the same virtual garment. The test also indicates that modification function plays an important role during the drawing and design procedure, which is also verified by participants' feedback.

Fig. 12 Examples of decoration tools

Generally, all participants found the prototype system convenient to generate garments for virtual avatars and would like to use it in an application. It provided an easy-to-use interface to implement their ideas for dressing 3D avatars. Participants desired two aspects of improvement to the current method and prototype system: one was to provide a multi-view drawing and modification function; the other was to simulate the animation of the dressing result.

5 Conclusion and future work

This paper presents an approach for interactively dressing 3D avatars based on a sketch interface. It allows users to create 3D garments for avatars by sketching garment contours around the avatar body. It is also demonstrates that the optimized garment mesh surface can be processed by applying some mesh processing methods such as subdivision and smoothing for possible further processing. The proposed method has been used and tested on 3D avatars from body scan data. The body scan data of human being has almost the same structure and similar body features. It is not suitable for some avatars in video games at this moment due to their various body features. We would like to extend the method in the future to make it more flexible for various types of avatar bodies. In the current implementation, we focus on the method for generating static garments for 3D avatars. This system does not take garment materials and animation into account. In applications, avatars should be able to be animated and move around in the virtual world. Since the generated garments are represented by uniform mesh surfaces, some cloth deformation methods could be applied to the garment mesh for animation and material simulation [5, 6, 23, 25, 31]. In future work, we would like to integrate cloth deformation and material simulation algorithms in the system and demonstrate animation in a virtual world. In addition, the clothing dragging method proposed in [14] could be applied to adjust the garment once it has been generated. Moreover, the system could be improved to allow users to draw and edit the garment in different views to create more complex garment styles. By combining 3D drawing/editing and flexible body constrain control, this method could be extended to allow users to dress avatar in video games in the future such as the Wii's MadWorld [38] etc. We also would like the system to display the 3D garment in real time as a user is sketching. The 3D garment displayed in real time before a user finishes sketching would be a suggested one by the system, which will give the user a reference with which he or she can compare the one imagined in his or her mind.

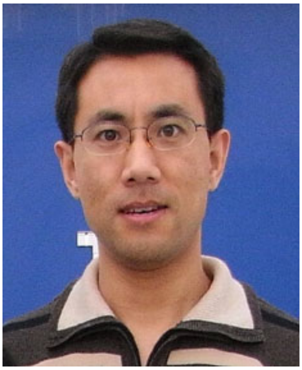
Acknowledgments This work is supported by Engineering and Physical Sciences Research Council, UK (EP/p501245) and partly supported by Projects of Beijing Municipal Commission of Education (KM200710005009 and PXM2009_014204_09_000154). The authors would like to thank all anonymous reviewers for their valuable comments. Thanks to Dr. Oliver G. B. Garrod for suggestions.

References

1. Arnulph F, Clemens G, Volker L, Andreas W (2003) Interaction-free dressing of virtual humans. Comput Graph 27(1):71–82
2. Bajaj CL, Coyle EJ, Lin KN (1996) Arbitrary topology shape reconstruction from planar cross sections. Graph Models Image Proc 58(6):524–543

3. Blanz V, Vetter T (1999) A morphable model for the synthesis of 3D faces. In SIGGRAPH'99. ACM Press, pp 187–194
4. Bourguignon D, Cani MP, Drettakis G (2002) Drawing for illustration and annotation in 3D. Comput Graphics Forum 20(3):114–122
5. Breen DE, House DH, Wozny MJ (1994) Predicting the drape of woven cloth using interacting particles. Comput Graph (SIGGRAPH'94 proceedings) pp 365–372
6. Chen M, Tang K (2010) A fully geometric approach for developable cloth deformation simulation. Vis Comput 26(6–8)853–863
7. CyberTown, http://www.cybertown.com/main_nsframes.html
8. DiPaola S, Collins D (1999) A 3d natural emulation design to virtual communities, Siggraph '99
9. DiPaola S, Collins D (2002) A 3D virtual environment for social telepresence. In Proc. Western Computer Graphics Symp. Vernon, BC, Canada
10. Fernando ONN, Adachi K, Duminduwardena U, Kawaguchi M, Cohen M (2006) Audio narrowcasting and privacy for multipresent avatars on workstations and mobile phones. IEICE Trans Inf Syst E89-D(1):73–87
11. Grand Theft Auto: San Andreas, Rockstar North Ltd. http://www.rockstarnorth.com/
12. Half-Life, Sierra Entertainment, Inc., http://www.sierragamers.com/
13. Igarashi T (2010) Computer graphics for all. Commun ACM 53(7):71–77
14. Igarashi T, Hughes J (2002) Clothing manipulation, 15th annual symposium on User Interface Software and Technology, ACM UIST'02, Paris, France, October 27–30, pp 91–1002
15. Igarashi T, Matsuoka S, Tanaka H (1999) Teddy: a sketching interface for 3D freeform design. Proc. 26th Conf. Computer Graphics and Interactive Techniques. ACM Press pp 409–416
16. Kinect. Microsoft Corporation. http://www.xbox.com/en-GB/kinect/
17. Kirkpatrick EM (1983) Chambers 20th century dictionary, p 84
18. Lawson CL (1977) Software for c1 surface interpolation, mathematical software III. Academic, New York, pp 161–194
19. Loop C (1987) Smooth subdivision surfaces based on triangles, Master's thesis, Department of Mathematics, University of Utah, Aug
20. Meyers D, Skinner S, Sloan K (1992) Surface from contours. ACM Trans Graph 11(3):228–258
21. Mullineux G, Robinson S (2007) Fairing point sets using curvature. Comput-Aided Des 39(1):27–34
22. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. J Comput Phys 79(1):12–49
23. Provot X (1995) Deformation constraints in a mass-spring model to describe rigide cloth behavior. Graphics interface'95 proceedings. AK Peters Ltd, pp 147–154
24. Ruppert J (1995) A Delaunay refinement algorithm for quality 2-dimensional mesh generation. J Algorithms 18(3):548–585
25. Sakagushi Y, Minoh M, Ikeda K (1991) PARTY: physical environment of artificial reality for dress simulation: a dynamically deformable model of dress. Trans Soc of Electronics, information and Communications, pp 25–32
26. Salomon D (1999) Computer graphics and geometric modelling. Springer, ISBN: 3879868209780387986821
27. Schroeder R (2002) The social life of avatars. Springer, New York
28. Second Life, Linden Lab. http://secondlife.com/
29. Shewchuk JR (1996) Triangle: engineering a 2D quality mesh generator and Delaunay triangulator in applied computational geometry: towards geometric engineering, lecture notes in computer science, vol. 1148. Springer-Verlag, Berlin, pp 203–222
30. Shewchuk JR (2002) Delaunay refinement algorithms for triangular mesh generation. Comput Geom Theory Appl 22(1–3):21–74
31. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rossl C, Seidel H (2004) Laplacian surface editing. Proceedings of the Eurographics/ACM SIGGRAPH symposium on Geometry Processing, vol. 71
32. Turquin E, Cani MP, Hughes J (2004) Sketching garments for virtual characters, Proc. Eurographics Workshop Sketch-Based Interfaces and Modeling, Eurographics, pp 175–182
33. Turquin E, Wither J, Boissieux L, Cani MP, Hughes J (2007) A sketch-based interface for clothing virtual characters. Comput Graph Appl 27(1):72–81
34. Vlasic D, Brand M, Pfister H, Popovic J (2005) Face transfer with multilinear models. ACM Trans Graph 24(3):426–433
35. Volino P, Thalmann NM (1997) Developing simulation techniques for an interactive clothing system. Proc. of Virtual Systems and MultiMedia, pp 109–118
36. Vollmer J, Mencl R, Müller H (1999) Improved Laplacian smoothing of noisy surface meshes. Comput Graphics Forum 18(3):131–138
37. Wang CCL, Wang Y, Yuen MMF (2003) Feature based 3D garment design through 2D sketches. Comput-Aided Des 35(7):659–672

38. Wii. http://uk.wii.com/
39. Yu H, Qin SF, Sun GM, Wright DK (2008) Automatic segmentation of scanned human bodies for building lod models. Industrial Informatics, 2008. INDIN 2008. 6th IEEE International Conference on Volume, Issue, 13–16 July, pp 1426–1431
40. Zorin D, Schroder P, Sweldens W (1996) Interpolating subdivision for meshes with arbitrary topology, SIGGRAPH 96, ACM, pp 189–192



Hui Yu received his Ph.D. degree in Graphics and Design Technology from Brunel University, UK. Dr. Yu is currently a post-doc researcher with University of Glasgow, UK. He received M.S degree in Image and Signal Processing. His research interests include, graphics modeling, dynamic facial expression modeling, recognition and perception.

Shengfeng Qin received his Ph.D. degree in Design Technology from University of Wales, UK. Dr. Qin is currently a senior lecturer with Brunel University, UK. He joined Brunel Design in 2001. He obtained his BSc and MSc degrees in China. He was an academic visiting scholar in 1996–97 within the Geometric Modelling group of the University of Birmingham UK after working as a Lecture from 1989 and an Associate Professor from 1994 in the East China Jiaotong University. Following the visiting, he took a Research Assistant post in 1998 at UWIC and subsequently joined the Manufacturing Systems Integration Research Institute (MSI), Loughborough University as a Postdoc Research Associate in 2000.

Guangmin Sun received his Ph.D. degree in Communication and Information Systems from Xidian University, China. Professor Sun received a BSc in electronic engineering from Beijing Institute of Technology, China, in 1982, an MS in communication and information systems from Nanjing University of Science and Technoloty, China, in 1991. He is currently a professor in the Department of Electronic Engineering at Beijing University of Technology. His research interests include neural networks and applications, pattern recognition, bioinformatics and signal and image processing.

David K. Wright is a Professor in the Department of Design, Brunel University, and head of Interaction Design Research Group. He gained his PhD in Physics at the University of Warwick. His teaching and research interests are in CAD, computer simulations of biomechanical systems; virtual prototype of products; as well as CSCW.