Computer-Aided Design 41 (2009) 614–625

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Interactive 3D garment design with constrained contour curves and style curves

Jin Wang [a,*], Guodong Lu [a], Weilong Li [a], Long Chen [b], Yoshiyuki Sakaguti [c]

[a] *State Key Lab. of CAD & CG, Zhejiang University, Hangzhou 310027, China*

[b] *College of Mechanical Engineering, University of Shanghai for Science and Technology, Shanghai, China*

[c] *Digital Fashion Ltd., Osaka, Japan*

ARTICLE INFO

Article history:
Received 13 November 2008
Accepted 16 April 2009

Keywords:
3D garment
Contour curves
Style curves
Constraints resolving
Surface modeling
Pattern design

ABSTRACT

This paper presents interactive techniques to design 3D garments conveniently and precisely with constrained contour curves and style curves. Contour curves, including silhouette curves and cross section curves, are used for garment surface modeling. Style curves, including seam lines, dart lines, notch lines and grain lines, are introduced for designing patterns on the triangular garment surfaces. Contour curves are extracted automatically from the boundaries of garment sub-meshes. The definitions and resolving rules of various constraints are introduced for editing the contour curves conveniently. Style curves are generated by projecting control points onto 3D garment triangular surfaces. In order to draw the style curves validly, some constraints are also introduced according to the craft requirements of pattern design. Furthermore, the effects of style curves in pattern flattening are analyzed, which can guide the designer to draw style curves more reasonably and enhance the pattern design quality. Finally, some examples are given to show that our methods can make the 3D garment design more flexible and friendly, and style curves can be applied into design patterns on 3D triangular surface for shoes, toys and so on.

1. Introduction

The purpose of this paper is to provide interactive techniques of 3D apparel design through multiple parametric curves, and it enables the stylist design surface shapes and patterns directly on a 3D human model more freely. Up to now, there has been three typical garment design methods, namely the *A*, *B*, *C* approaches shown in Fig. 1. The traditional 2D CAD systems provide such 2D grading tools to generate patterns of different sizes from the basic pattern set, and then use them to make garments, as path *A* shown in Fig. 1. Along with the development of garment simulation techniques [1–3], some commercial CAD systems test the garment design results by assembling 2D patterns and draping them on a virtual human body [4–10], as path *B* shown in Fig. 1. Such solutions are not intuitive enough and need the designers to have rich experiences and accomplished skills. In order to make the design process more intuitive, Volino et al. [11] provided an interactive design environment to edit the patterns either in 2D or 3D with immediate preview of the garment draping result. However, it still needs to modify the 2D patterns many times for generating a fit 3D garment.

With the development of 3D laser scan and surface reconstruction technology, more and more studies focus on designing a garment directly on a 3D scanned body, and 3D CAD technology is gradually diffusing into the garment design and manufacturing applications. The 3D approach consists of several elements, which include 3D body modeling, 3D garment generation, interactive 3D garment surface modeling and pattern design, and 3D/2D pattern conversion, as path *C* shown in Fig. 1. Compared to other elements, few studies have focused on providing effective tools for 3D garment interactive design.

Due to the convenience of sketch based interactive techniques, recently some papers turned to designing garments by sketches [12–14]. However, it only suits conceptual design, not the apparel industry.

For 3D garment design in the apparel industry, there are some problems that need to be solved.

(1) How to design 3D garments more effectively. For a complex garment surface design, it needs effective tools to edit the outline and cross section shape directly and easily. For a complex pattern design, it is necessary to draw not only seam lines, dart lines but also other curves according to the craft requirements.
(2) How to design 3D garments validly and reasonably. For the apparel industry, constraints have to be considered for convenient editing and satisfying some craft requirements. Moreover, the users need to be guided to tailor the pattern reasonably, and thus enhance the pattern quality.
(3) How to make the interactive tools fit the process of garment design, which includes outline modeling and inner region tailoring. It is meaningful in the apparel industry to make the

* Corresponding author. Tel.: +86 571 87951273; fax: +86 571 56891700.
E-mail address: dwjcom@zju.edu.cn (J. Wang).

0010-4485/$ – see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cad.2009.04.009

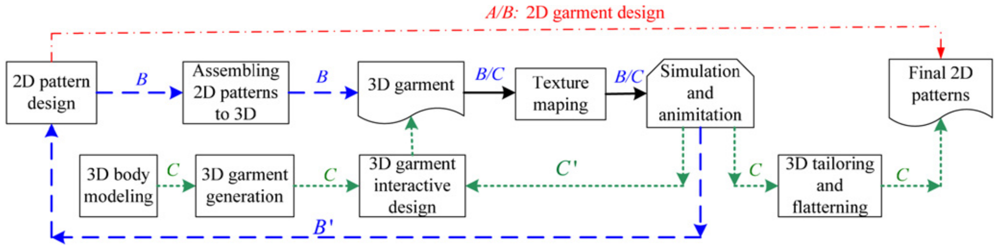


Fig. 1. Typical garment design methods.

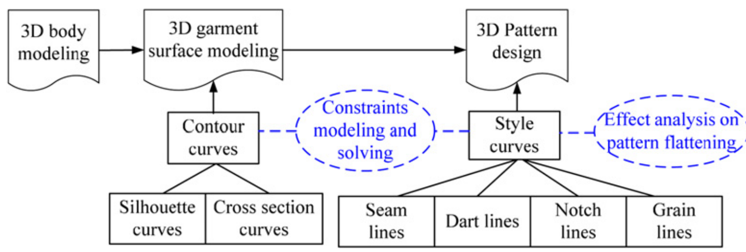


Fig. 2. 3D interactive garment design with constrained contour curves and style curves.

design process separate and suitable for the cooperation of fashion designers and pattern makers.

This paper introduces parametric contour curves and style curves to make the 3D garment design direct, precise and suitable for the cooperation of fashion designers and pattern makers, as shown in Fig. 2. Contour curves, including silhouette curves and cross section curves, are used to model the garment surfaces. Style curves, including seam lines, dart lines, notch lines and grain lines, are used to tailor 3D patterns. Constraints are introduced to edit contour curves conveniently and make style curves satisfy some craft requirements of pattern design. Effect analysis of style curves on pattern flattening is introduced to guide the designer to draw style curves more reasonably, and thus enhance the quality of pattern design.

The contribution of this paper is listed as follows.

(1) Classify 3D garment design into surface modeling and pattern tailoring with contour curves and style curves for the cooperation of fashion designers and pattern makers.
(2) Propose a lot of detailed curves to design the 3D garment more effectively. Especially the multiple style curves can be used to tailor complex 3D patterns on any triangular surface.
(3) Introduce constraints and effect analysis to model the surface conveniently, tailor patterns reasonably and fit craft requirements.

The structure of this paper is as follows: Section 2 reviews related work. In Section 3, contour curves are automatically extracted from the boundaries of garment submeshes. In Section 4, constraint definitions and solving methods are introduced for modeling the garment surface conveniently. Section 5 describes the definitions, generation methods and drawing constraints of style curves for 3D pattern design. Some typical pattern results with different style curves are listed in this section. In Section 6, some design examples are presented, and finally the paper is summarized in Section 7.

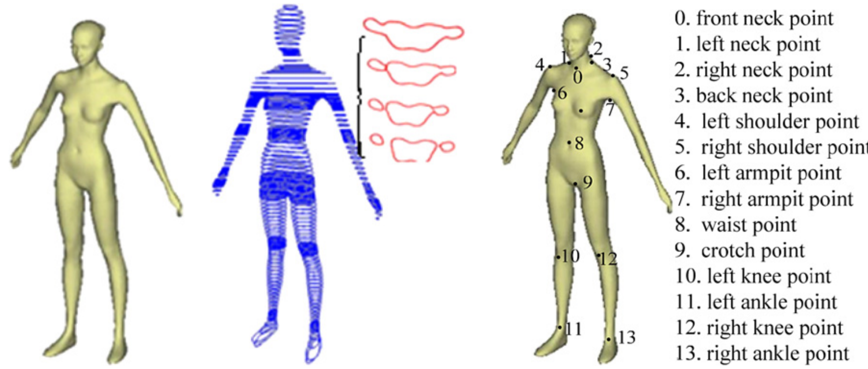
2. Related works

Some works have been done for generating garment surface on a 3D body. In Hinds and McCartney's work [15,16], the garment is represented as a collection of panels offset from the body surface, and the garment panels are constructed around a static mannequin body. Kim et al. [17,18] divided the body model into four panels and adopted a stereovision based algorithm to generate the garment surface, but it needs to draw appropriate grids on the body first as the reference for constructing the garment mesh. Based on the body features, Wang [19] specified a garment sketch through a 2D stroke and then used a subdivision method to refine the garment mesh. Wang [20] separated the laser scanned body into six parts, and connected the human body slice vertices intersecting with each part by horizontal cutting plane to reconstruct the body surface.

Some simple interactive tools were proposed to edit the garment surface. Wang [21] adopted developability-preserved FFD methods to deform the generated garment surfaces. Parametric curves such as sine curves were used to control the bottom garment shape more easily in the works of Kim et al. [18]. Wang et al. [21,22] proposed four simple modification tools for the generated garment surface by 2D sketches, which were mesh painting, mesh extrusion, mesh cutting and mesh partition.

Along with the development of sketch interactive techniques, some free-form modeling methods are proposed. The SKETCH system and teddy system can rapidly construct approximate shapes via 2D closed stroke. Turquin et al. [12] introduced this technique into garment generation. It allows the users to sketch garment contours directly onto a 3D view of a mannequin. A distance field around the mannequin is then used to generate a 3D surface. However, it does not generate patterns for sewing, and it only suits virtual characters. To overcome this shortage, Decaudin et al. [14] added seam lines during sketching garment contours to generate patterns, and proposed some methods to make the patterns developable.

Pattern flattening is another important work for 3D garment design, and many works have been done to enhance its quality. McCartney et al. [23] flattened the triangulated surfaces incorporating darts and gussets to reduce the strain energy. Kim et al. [18] added darts on the usual position based on the design experience, and it was hard to draw on the part with high curvature exactly. Wang et al. [24] depicted the surface elastic deformation energy by a color graph, and then determined a surface cutting line that passes through the maximum energy point along which the descent of the energy is the slowest. In paper [25], Wang et al. computed the shortest path from a node to the surface boundary for automatically generating dart lines. However, the shape of automatically generated darts is simple, not smooth and hard to be



(a) Body. (b) Slices. (c) Main feature points.

Fig. 3. Human feature point recognition.

edited. Wang et al. [26] also took material properties into consideration to reduce the deformation in surface flattening. Besides inserting darts, Wang et al. [20] and Decaudin et al. [14] proposed some approaches to make the garment surface more developable, and thus, to enhance the pattern flattening quality.

According to these works, many efforts have been done on 3D garment generation and pattern flattening. Simple tools are proposed for interactive design, and sketches turn out to be the focus of study as a convenient interactive tool. However, sketches are hard for acquiring smooth and precise shapes, and unsuitable for iterative modification. Most studies do not classify the 3D garment design into surface modeling and pattern tailoring for the cooperation of fashion designers and pattern makers, for example, Turquin et al. ignores inner region tailoring and Decaudin et al. sketch the contour lines and tailor lines together. Few works focused on the constraints in surface modeling and pattern tailoring, and most studies only focused on analyzing the effect of dart lines.

Parametric curves, especially the free-form curves, are also suitable for free interaction, and can be easily edited into precise and complex shapes. Thus, this paper proposes an interactive solution for 3D garment design, including definition, generation, constraints modeling and solving, effect analysis of parametric curves.

3. 3D garment surface generation and contour curve automatic extraction

3.1. Recognition of human feature points

Human feature points are the base of automatic 3D garment generation. After a series of cross sections are acquired by intersecting the body with horizontal cutting planes from head to foot, feature points can be searched by judging the number and geometric shape information of cross sections [13]. Taking the body height and the position scale of feature points as reference, the recognition process can be speeded up. In the dense region of feature points, we add the number of cutting planes, as shown in Fig. 3(b). Fig. 3(c) shows about 14 recognized main feature points. Actually, for most of the body scan systems, it can provide more than forty feature points together with the scanned body data.

3.2. Body reconstruction and 3D garment surface generation

Piecewise techniques are used to reconstruct the body more easily. Based on the feature points, the body is divided into several parts, such as legs, trunk, the front and back chests, the front and back under-necks, the left and right shoulders, the left and right arms, as shown in Fig. 4(a). Each part has simple topology, and it is easy to get a series of slice vertices by cutting the body with planes. A series of small meshes can be formed by connecting these slice vertices, and are called sub-meshes of the garment surface. In order to construct the mesh more evenly, we adjust the directions of cutting planes adaptively, as shown in Fig. 4(b). Since most of the bodies and garments are almost symmetric, we make the body surface symmetric via two steps. First, let the posture of the model be as symmetric as possible. Second, we make the left and right corresponding slice vertices symmetric along the center line that is connected by neck and crotch feature points F_{p1}, F_{p2}, as shown in Fig. 4 (b). Here, $x_{cen} = F'_{p1.x} = F'_{p2.x} = (F_{p1.x} + F_{p2.x})/2$, $P'_{li.x} = (x_{cen} - (P_{ri.x} - P_{li.x})/2)$, $P'_{li.y} = (P_{li.y} + P_{ri.y})/2$, $P'_{li.z} = (P_{li.z} + P_{ri.z})/2$. In this paper, we ignore the special case of obvious asymmetric body.

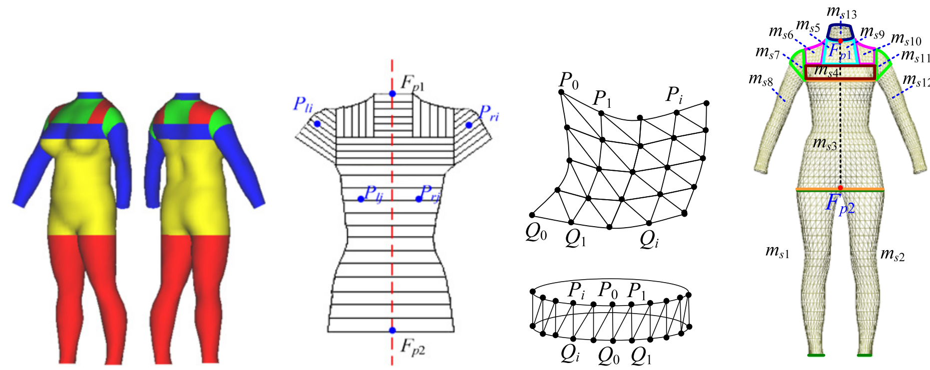
As shown in Fig. 4(c), connect the slice vertices $\{\{P_0, P_1 \ldots P_i \ldots\} \ldots \{Q_0, Q_1 \ldots Q_i \ldots\}\}$ to construct a triangular mesh $M = \{P, E, T\}$. Here $P = \{P_i\}$, $E = \{E_k\}$, $T = \{T_j\}$, denote the vertices, edges, and triangles sets, respectively. Their topology linkage information are defined as follows, $P_i = \{\{AE_i\}, \{AT_i\}\}$, $T_j = \{(P_{j0}, P_{j1}, P_{j2}), (E_{j0}, E_{j1}, E_{j2}), \{AT_j\}\}$, $E_k = \{(P_{k0}, P_{k1}), \{AE_k\}, \{AT_k\}\}$, where $\{AE_i\}$, $\{AE_k\}$ are the adjacent edge sets of P_i and E_k, respectively; $\{AT_i\}$, $\{AT_j\}$, $\{AT_k\}$ are the adjacent triangle sets of P_i, T_j, E_k, respectively.

Let M_s, M denote the sub-meshes and whole mesh of the garment surface. Merge the vertices and triangles in $\{M_{si}\}$, cull redundant vertices and reset connect relations, and then M is constructed, as shown in Fig. 4(d). Offset the vertices in M along their normal V_p, the garment surface can have margins δ with the body. That is $M' = \{T, E, P'\}$, where $P' = \{p'_i\}$, $P'_i = P_i + V_p \cdot \delta$. Also we can make the garment to have uneven margin by controlling the δ. Since the sleeves have little connection with the arm shape, and are always constructed by parametric models [27], here we do not merge the sub-meshes of arms and construct a sleeveless garment only, as shown in Fig. 5(b).

3.3. Automatic extraction of contour curves

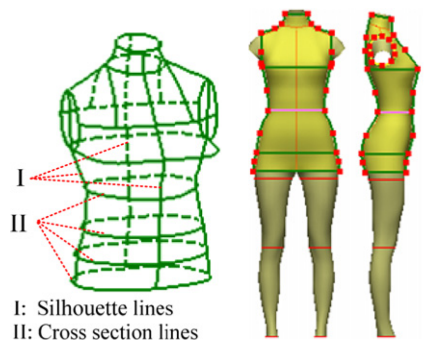
In order to model the garment surface effectively, contour curves are classified into silhouette curves and cross section curves, as shown in Fig. 5(a). Silhouette curves are extracted from the boundaries of garment sub-meshes, and are used to control the outline of the garment. Cross section curves are generated from cutting the garment surface with planes at feature positions, such as breast, waist, hip and so on, and are used to model the cross section shape of the garment effectively.

Free-form curves are used to represent the extracted line segments for convenient editing, as shown in Fig. 5(b). Silhouette curves can be represented as B-Spline curves, as listed in Eq. (1). However, the Ferguson curve is hard for acquiring a conical shape



(a) Body pieces. (b) Cutting plane directions. (c) Sub-meshes. (d) Whole body mesh.

Fig. 4. Garment surface construction based on piecewise technique.



(a) Sleeveless garment contour curves. (b) Sleeveless garment surface and contour curves.

Fig. 5. Extraction of the contour curves on sleeveless garments.

precisely, and the C-Spline has a similar property to the B-Spline except that it can represent the conical shape precisely [28–30], as listed in Eq. (2). So for cross section curves, we can adopt C-Splines to represent them.

$$P_i(u) = \frac{1}{6}\left[u^3\ u^2\ u\ \ 1\right]\begin{bmatrix}-1 & 3 & -3 & 1\\ 3 & -6 & 3 & 0\\ -3 & 0 & 3 & 0\\ 1 & 4 & 1 & 0\end{bmatrix}\begin{bmatrix}V_i\\ V_{i+1}\\ V_{i+2}\\ V_{i+3}\end{bmatrix} \tag{1}$$

$$\begin{aligned}P_i(u) \;&=\; \frac{1}{2\alpha(1-C_\alpha)}\begin{bmatrix}C_\alpha & -(1+2C_\alpha) & 2+C_\alpha & 1\\ -S_\alpha & 2S_\alpha & -S_\alpha & 0\\ -1 & 1+2C_\alpha & -(1+2C_\alpha) & 1\\ \alpha & -2\alpha C_\alpha & \alpha & 0\end{bmatrix}\\ &\times\begin{bmatrix}V_i\\ V_{i+1}\\ V_{i+2}\\ V_{i+3}\end{bmatrix}\quad C_\alpha = \cos(\alpha), S_\alpha = \sin(\alpha),\\ &0 \le \alpha \le \pi, 0 \le u \le \alpha, (i = 0, 1, \ldots, n-1).\end{aligned} \tag{2}$$

The process of generating the sleeve surface and contour curves is similar, and its details are presented in our team member's paper [27].

4. 3D garment surface modeling via contour curves

4.1. Constraint definition and description of editing contour curves

Constraint modeling and resolving are problems in editing curves conveniently. According to the features of contour curve editing, 10 major constraint types are defined in our paper [31], which are the same point constraint (C_1), the same plane constraint (C_2), the symmetric constraint (C_3), the self-symmetric constraint (C_4), the movement follow constraint (C_5), the precise size constraint (C_6), the fuzzy size constraint (C_7), the mutual orientation constraint (C_8), the collision detection constraint (C_9) and the self-intersection detection constraint (C_{10}) as shown in Fig. 6. Note that, if we design an asymmetric garment on an asymmetric body, C_3, C_4 are not adaptable.

Define a constraint as $C_i = \{T_p, (e_1, e_2, \ldots e_i \ldots), H_d\}$, where T_p denotes the constraint type, $(e_1, e_2, \ldots e_i \ldots)$ denotes the set of the constrained geometric elements, *CGE* for short, H_d denotes constraint handling methods or rules. Previous 10 constraints can be described as listed in Table 1 [31]. In this paper, we focus on how to construct the constraints graph automatically and more detailed constraints resolving methods. Then we set these constraints solutions in the program as a preparation, and it can make the interaction editing process convenient, which is important for commercial software.

4.2. Constraint graph construction for contour curves

The moving correlation between the contour curves can be reflected by geometric constraint satisfaction and resolving. Graph based constraint resolving has high efficiency, and it can handle over or under constraints. Thus, this method is used widely. According to the driven relation between constraints and curve elements, constraints are classified into drive and restraint types. Thus, a constraint graph can be constructed conveniently.

Drive constraints, *DC* for short, drive the movement of geometric elements. Restraint constraints, *RC* for short, restrain geometric elements from moving out of range. If we want to enlarge the armhole range for padding or adjust the shoulder shape, we edit relative points around L_{s3}, as shown in Fig. 7. While L_{s3} is moving, it drives R_{s3} moving by C_3, and in this case, C_3 is a *DC*. During the moving process of R_{s3}, it is restrained by C_6, C_8, C_9 from moving out of range. Note that, whether a constraint is *DC* or *RC* lies on the concrete constraint case.

According to our study, we proposed some rules to construct the constraint graph conveniently.

Rule 1. Constructs the *DC* of an element firstly, and then constructs the *RC*.

Rule 2. Constructs all the *RCs* of the current element first, then we search other elements driven by current element via *DC* and construct constraint graphs on these elements.

According to these rules, as shown in Fig. 7, when the control point L_{s3} is moving, the constructing process of constraint graph can be listed as follows.

(1) First, construct the *RCs* of L_{s3}, that is C_6, C_8, C_9. Only *RCs* are resolved, L_{s3} can be restrained in a reasonable position.

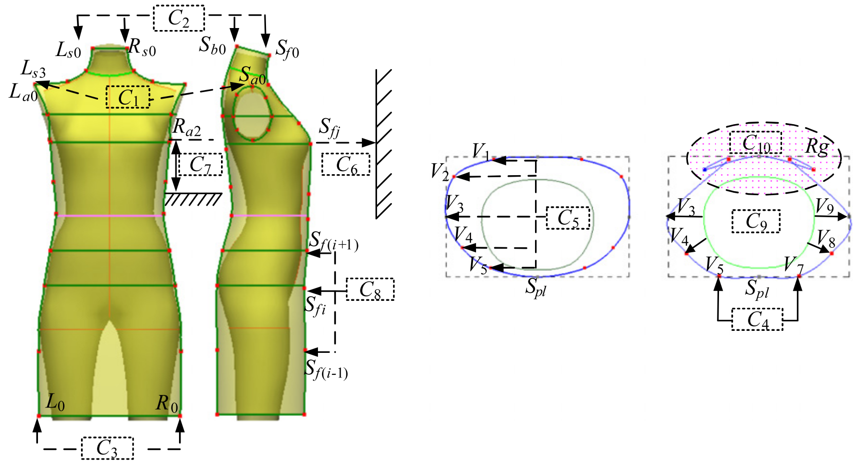


Fig. 6. Constraint definition of editing contour curves.

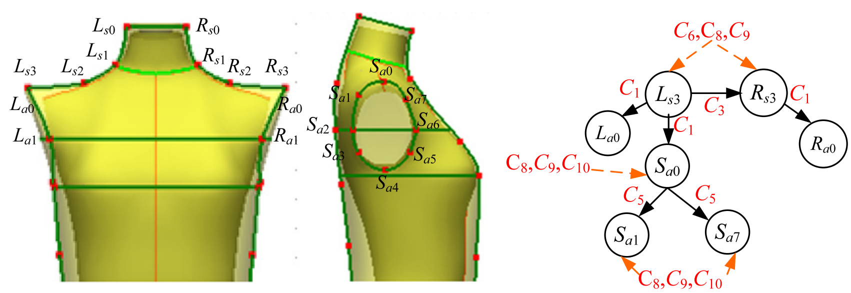


Fig. 7. Constraint graph construction for contour curves.

Table 1
Definition and description of ten constraints in editing contour curves.

T_p	Name	$CGE(\ldots e_i, \ldots)$	Description (H_d)
C_1	Same point	(L_{s3}, L_{a0}, S_{a0})	All the *CGE* have the same coordinate values.
C_2	Same plane	$(L_{s0}, R_{s0}, S_{f0}, S_{b0})$	All the *CGE* are on a same plane.
C_3	Symmetric	(L_0, R_0)	The *CGE* are symmetric about one plane and belong to different entities.
C_4	Self-symmetric	(V_5, V_7)	The *CGE* are symmetric about one plane and belong to the same entity.
C_5	Movement follow	$(V_1, V_2, V_3, V_4, V_5)$	When a main element is moving, other *CGE* follow the movement by scale. As shown in Fig. 6, in order to move the cross section evenly. When a point V_3 is moving, points V_1,V_2,V_4,V_5 moves along with V_3 by scale.
C_6	Precise size	(S_{fj})	The *CGE* should not move out of range in *x*, *y*, *z* directions precisely. As shown in Fig. 6, the control point S_{fj} should not exceed a maximum *z* coordinate and also cannot keep the garment sizes out of range after the garment surface is deformed by S_{fj} moving.
C_7	Fuzzy size	(R_{a2})	The *CGE* should not exceed a specified relative range. As shown in Fig. 6, the control point R_{a2} should not be moved below the middle of the bust and waist, such a range is a relative value, so we call it a fuzzy size constraint.
C_8	Mutual orientation	$(S_{f(i-1)}, S_{fi}, S_{f(i+1)})$	Each *CGE* should not move over each other in one or several directions. As shown in Fig. 6, control points $S_{f(i-1)}$, S_{fi}, $S_{f(i+1)}$ should be monotone in the *y* direction
C_9	Collision detection	$(\ldots V_3, V_4 \ldots)$	The *CGE* should not collide with the reference objects, such as body mesh, or body cross section curves, or body silhouette curves.
C_{10}	Self-intersection detection	(S_{pl})	The *CGE* cannot intersect with itself. As shown in Fig. 6, the garment cross section S_{pl} intersects at the shade region R_g, such cases should be avoided.

(2) L_{s3} drives L_{a0} through C_1, drives R_{s3} through C_3.
(3) After R_{s3} is restrained by C_6, C_8, C_9 in a reasonable range, it drives R_{a0} through C_1.
(4) Since L_{s3} and S_{a0} belong to an identical point, L_{s3} drives S_{a0} through C_1.
(5) S_{a0} is restrained by C_8, C_9, C_{10} first, and then drives S_{a1}, S_{a7} moving by scale through C_5.
(6) S_{a1}, S_{a7} are restrained by C_8, C_9, C_{10}.

4.3. Constraint resolving of contour curves and garment surface modeling

Based on the constraint graph, starting from the first element, pushing the next element linked by *DC* recursively, a stack for constraint resolving can be quickly established. For each popped element, we search all the *RC* and resolve them one by one. However, an element is always restrained by several constraints. Thus constraint classification and resolving rules have to be established.

Constraints are classified into 3 categories, namely topology constraints C_{tp1}, correlation constraints C_{tp2}, and dimensional constraints C_{tp3}. C_1, C_2 belong to C_{tp1}, C_3, C_4, C_5, C_8, C_9, C_{10} belongs to C_{tp2}, and C_6, C_7 belong to C_{tp3}.

Depending on the severity degree that customers feel through the visual impression when the constraint is crashed, a strict level for three constraint types is supposed as follows:

***Strict level* 1:** $C_{tp1} > C_{tp2} > C_{tp3}$.

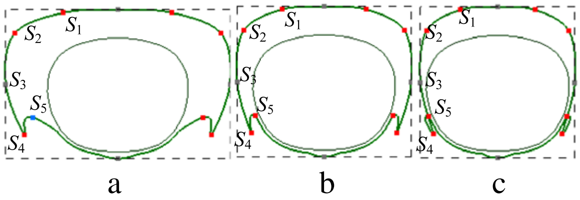
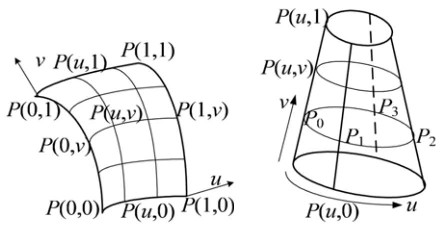


Fig. 8. Release the constraints with lower strict level.



(a) u, v definition of mesh I.

(b) u, v definition of mesh II.

Fig. 9. The Coons surface generation.

Here, C_{tp1} maintains the topology correlations of the contour curves, and if it is broken, the geometric system will be crashed; so it is the most strict constraint type. C_{tp2} maintains the position correlations between the control points. Then the visual impression is not very bad if they are broken in a small region, and it may be recovered again according to C_{tp1} in the later editing process; so it is more relaxed than C_{tp1}. If C_{tp3} is broken, it will not effect the topology and position correlation, and is hard to be observed when it is broken in a small region. We also can correct it via dimension scale, so it is more relaxed than C_{tp1} and C_{tp2}.

Based on ***strict level*** **1**, taking the program debugging and user's opinions as references, a detailed strict level of previous ten constraints is supposed as follows.

Strict level **2**: $C_1 > C_2 > C_9 > C_{10} > C_4 > C_3 > C_8 > C_6 > C_7 > C_5$.

Here, C_5 is a special case. It only drives relevant control points moving proportionally, actually, how long other elements moves or whether other elements can move will not break the correlations between the control points. Thus, even it belongs to C_{tp2}, we regard it as the most relaxed constraint.

The stricter constraint should be resolved later to make sure that it is first satisfied. The sequence of constraint resolving is as follows.

Resolve sequence **1**: $DC > RC$

***Resolve sequence* 2:** $C_5 > C_7 > C_6 > C_8 > C_3 > C_4 > C_{10} > C_9 > C_2 > C_1$

Note that, ***Resolve sequences*** **1** and **2** are used separately. For each element, DC is firstly resolved, and if an element has several RCs, then ***Resolve sequence*** **2** is applied.

However, if no solution results or the result affects the smooth operation, we release the constraints with a lower strict level, and remind the user to adjust the result, as presented in our paper [31]. For example, as shown in Fig. 8(a), control points S_1, S_2, S_3, S_4, S_5 are moved into a special case, and S_5 will be constrained during S_1, S_2, S_4, S_5 following the movement of S_3, as shown in Fig. 8(b). If S_1, S_2, S_4 continue to move self-intersection may occur, as shown in Fig. 8(c). In order to keep the operating continuity, we ignore the C_{10} constraint and remind the user to adjust the final result.

Constraints can be resolved effectively after the resolve sequence is proposed, and then the contour curves are deformed conveniently. Since contour curves are the boundaries of garment sub-meshes, and each sub-mesh has four boundaries in a sleeveless

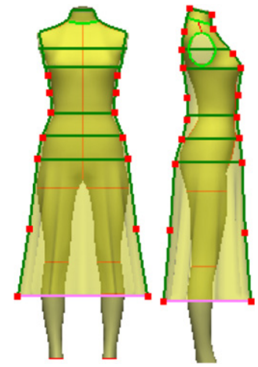


Fig. 10. Garment surface modeling by editing silhouette curves.

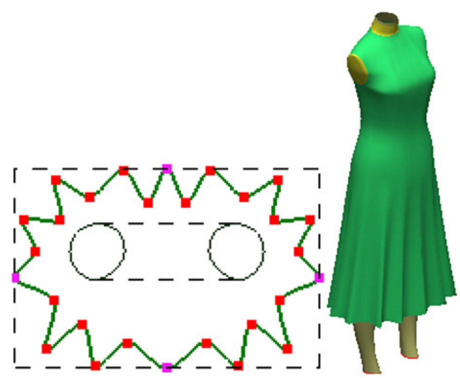


Fig. 11. Simple wrinkle shape modeling of skirt by editing cross section curves.

garment, it can be easily deformed through bilinear Coons interpolation method described in Eq. (3). Fig. 9(a), (b) shows two types of u, v definitions for different sub-meshes. Such as the trunk sub-mesh of the garment, below breast, takes category II as reference for deformation. Figs. 10 and 11 show the result of the garment surface modeled by silhouette and cross section curves. The sleeve surface modeling, has been introduced in detail in paper [27], including the process of interpolation of triangular surfaces, as M_{s7}, M_{s11} shown in Fig. 4(d).

$$P(u,v) = -[-1, u, 1-u]\begin{bmatrix} 0 & P(u,0) & P(u,1) \\ P(0,v) & P(0,0) & P(0,1) \\ P(1,v) & P(1,0) & P(1,1) \end{bmatrix}\begin{bmatrix} -1 \\ v \\ 1-v \end{bmatrix} = -M_u B M_v, \quad u, v \in [0,1] \tag{3}$$

where, $P(u,v)$ is an arbitrary point in u, v parameterized mesh, $P(u,0)$, $P(u,1)$, $P(0,v)$, $P(1,v)$ are four boundaries of the parameterized mesh and $P(0,0)$, $P(0,1)$, $P(0,1)$, $P(1,1)$ are four corner points.

After the surface is deformed, it needs to undergo collision detecting with the mesh vertices, and here we do not introduce this process. If the body is changed locally, we can update the corresponding contour curves via dimensional driven or collision detection, and then update the garment surface via bilinear Coons interpolation and vertices collision detection.

5. Style curves based pattern design on 3D triangular surface

5.1. Description and definition of style curves on 3D triangular surface

Let the 3D garment surface be S_v. In order to make the pattern design more flexible on S_v, we proposed a series of style curves with different types, such as straight seam lines C_L, spline seam lines F_L, dart lines D_L, grain Lines G_L and notch Lines N_L, as shown in Figs. 12 and 13. These style curves are generated on S_v, not on a free space. Their endpoints are mapped onto corresponding

Table 2
Description of the style curves.

Types	Description	Examples in Fig. 12	Application
C_L	Seam curves, drawn on the S_v with straight lines	C_{L0}, C_{L1}	Make up of the boundaries of P_{n3d}.
F_L	Seam curves, drawn on the S_v with spline curve	F_{L0}, F_{L1}, F_{L2}, F_{L3}	Make up of the boundaries of P_{n3d}.
D_L	Dart curves, drawn on the S_v with C_L or F_L	D_{L0}	Add on the high curvature part to reduce the deformation in pattern flattening [10,19,20].
G_L	Grain curves, drawn on the S_v with an arrow	G_{L0}	Denotes the lengthwise direction of the pattern. It restrains the process of pattern flattening to make the specified pattern boundary parallel with the grain lines.
N_L	Notch curves, drawn on the S_v as the shape shown in Fig. 13(e)	$N_{L0}, N_{L1}, \ldots, N_{L5}$	Label the matching position for sewing patterns.

Table 3
Definition of the style curves.

Types	Definition	Sign	Parameters' meaning
C_L	$(C_p, C_f, E_g, I_p, I_f, ID, IDA)$	Fig. 13(a)	$C_p = \{C_{p0}, C_{p1}\}$, the endpoints of C_L. $C_f = \{C_{f0}, C_{f1}\}$, the triangles *ID* that C_p locates. $E_g = \{E_{gi}\}$, edges sets that C_L goes through. $I_p = \{I_{pi}\}$, intersection points of C_L and E_g. $I_f = \{I_{fi}\}$, *ID* of the triangles that I_p locates. *ID*, ID of C_L. *IDA*, *ID* sets of the linked style curves.
F_L	$(S_{pl}, \{S_{fi}\}, \{C_{Li}\}, ID, IDA)$	Fig. 13(b)	S_{pl}, a cubic spline contains a series of control points $\{S_{pi}\}$. $\{S_{fi}\}$, the triangles *ID* that $\{S_{pi}\}$ locates. $\{C_{Li}\}$, is the C_L sets which are made up of F_L. *ID*, *ID* of F_L. *IDA*, *ID* sets of the linked style curves.
D_L	$(C_L\|F_L, T_p)$	Fig. 13(c)	$C_L\|F_L$, geometric elements of D_L. T_p, mark it a dart line.
G_L	$(\{CL_i\}, ID)$	Fig. 13(d)	$\{CL_i\}$, a series of C_L that make up the arrow shape. *ID*, *ID* of G_L.
N_L	$(\{C_{Li}, C_{Li\_2d}\}, ID, ID\_on)$	Fig. 13(e)	$\{C_{Li}, C_{Li\_2d}\}$, a series of C_L that make up its shape on 3D and 2D. $C_{Li\_2d} = \{C_{p\_2d}, C_f, E_g, I_{p\_2d}, I_f, ID, IDA\}$, denotes a C_L generated on 2D meshes. *ID*, *ID* of N_L. *ID_on*, *ID* of C_L or F_L that N_L locates.

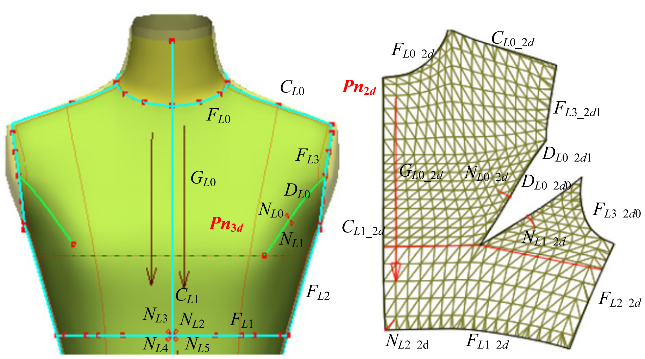


Fig. 12. Style curves with different types.

triangles, and their track path will intersect mesh edges of S_v. Thus, they have some related information, such as intersection points, edges and triangles that needs to be recorded. Their description and definition are listed in Tables 2 and 3, as shown in Figs. 14, 16 and 17.

5.2. Generation of style curves on 3D triangular surface

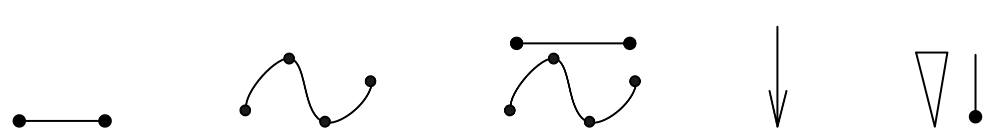
According to the definition of C_L, there are three main steps in generating C_L on S_v. First, acquire $\{C_{p0}, C_{p1}\}$, $\{C_{f0}, C_{f1}\}$ according to the 2D inputted points. Second, utilize C_p, C_f to construct a plane for intersecting the mesh edges of S_v and getting $\{I_{pi}\}$, $\{E_{gi}\}$, $\{I_{fi}\}$. Third, organize the information to form a C_L. As shown in Fig. 14, the generating process is listed as follows.

(1) Pick the triangle *ID* C_{f0}, C_{f1} on S_v according to the two inputted 2D points P_1, P_2 and then convert P_1, P_2 to 3D points V_1, V_2 using OpenGL function [32].
(2) Project V_1, V_2 onto S_v along the view direction V_{nc} and get the mapped point C_{p1}, C_{p2}.
(3) Let the normal of C_{f0}, C_{f1} be $\boldsymbol{V}_{n1}$, $\boldsymbol{V}_{n2}$, Taking $(C_{p0} - C_{p1})$ as path and $V_n = (\boldsymbol{V}_{n1} + \boldsymbol{V}_{n2})/2 \times (C_{p0} - C_{p1})$ as normal, we construct a plane P_{ln} to cut S_v, as shown in Fig. 15(a).
(4) Collect all the mesh edges $\{E_{gi}\}$ that intersect with P_{ln}, and compute all the intersection points $\{I_{pi}\}$.
(5) Start from C_{f0}, according to the topology linkage information of mesh edges, we collect all the $\{I_{pi}\}$ and $\{E_{gi}\}$ until an edge E_{gi} connecting with C_{f1}. Two adjacent edges E_{gi}, E_{gi+1} can get a triangle I_{fi}, and thus I_p, E_g, I_f information of C_L can be quickly acquired.
(6) Construct C_L by organizing the information of $\{C_{p0}, C_{p1}\}$, $\{C_{f0}, C_{f1}\}$, I_p, I_f, E_g.

In step (3), we also can let $V_n = V_{nc} \times (C_{p0} - C_{p1})$ to specify another cutting plane $P'_{ln.}$, where V_{nc} is the view direction. As shown in Fig. 15(b), (c), C_{L1} takes plane P_{ln} while C_{L0}, C_{L2} adopt $P'_{ln.}$ in the process of generation. We can find that C_{L0}, C_{L2} seem 'straight' along their respective view directions, and their paths are different even if their endpoints and triangles are identical. However, C_{L1} is unique if only its endpoints and triangles are identical in spite of the view direction.

As shown in Fig. 16, an F_L can be generated by the following steps:

(1) Collect a series of points $\{S_{pi}\}$ which are mapped onto S_v and their located triangles $\{S_{fi}\}$.
(2) Construct a spline S_{pl} using $\{S_{pi}\}$, and discretize a series of 3D points $\{p_i\}$ as the input points for $\{C_{Li}\}$.
(3) Project $\{p_i\}$ onto S_v to get the mapped points and triangles as $\{C_{pi}\}$ and $\{C_{fi}\}$ of $\{C_{Li}\}$.



(a) C_L. (b) F_L. (c) Dart. (d) Grain line. (e) Notch.

Fig. 13. Sign of style curves.

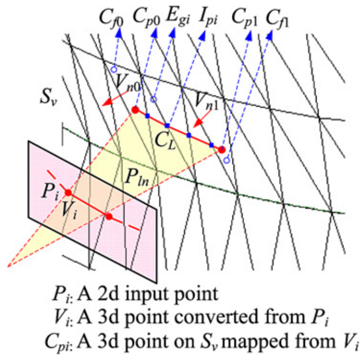


Fig. 14. Definition and generation of C_L.

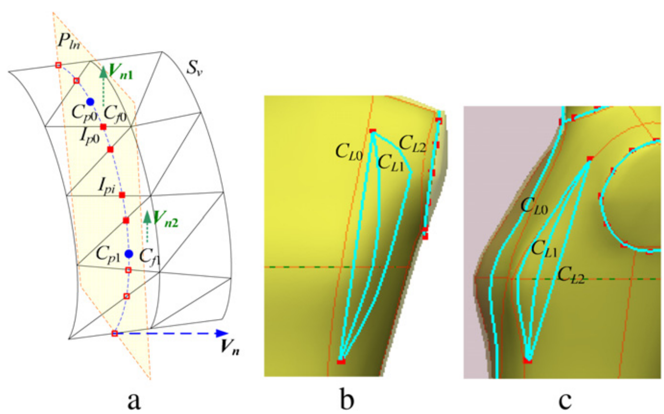


Fig. 15. C_L generation adopting different cutting planes.

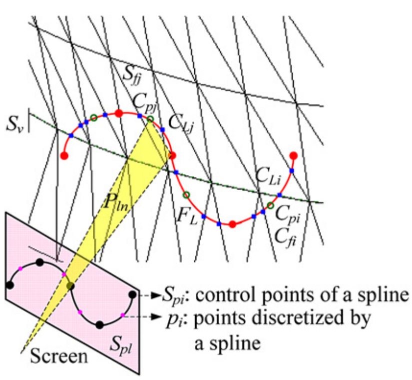


Fig. 16. Definition and generation of F_L.

(4) Generate a series of C_L by the previous method and convert them into $\{C_{Li}\}$ of F_L.
(5) Construct F_L by organizing the information of S_{pl}, $\{S_{fi}\}$, $\{C_{Li}\}$.

D_L is interactively drawn on high curvature parts using the generation method of C_L or F_L, such as the breast or shoulder, and finally label it D_L. We use color graph to indicate the curvature, and we introduce it in Section 5.4. D_L does not cut the mesh during its drawing and generating until in the pattern flattening process [10,19,20], as the 2D pattern $\boldsymbol{P}_{n2d}$ shown in Fig. 12.

As shown in Fig. 17, G_L is an arrow shape and can be generated by several C_L in a pattern region as follows.

(1) Take the inputted control points C_{p0}, C_{p1} and their locating triangles as reference, we generate a C_L to form the trunk of the arrow shape on S_v. Here, we use $\boldsymbol{V}_n = \boldsymbol{V}_{nc} \times (C_{p0} - C_{p1})$ to construct the cutting plane, and make the C_L seem straight along the view direction.

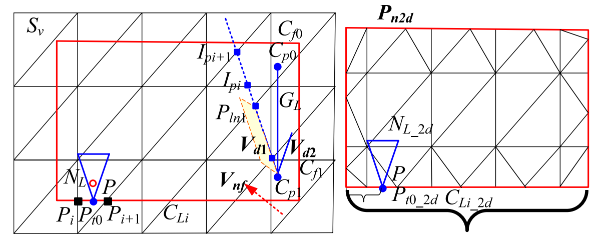


Fig. 17. Definition and generation of G_L and N_L.

(2) Let the normal of C_{f1} be $\boldsymbol{V}_{nf}$. Taking $\boldsymbol{V}_{nf}$ as axis, we rotate the path $(C_{p0} - C_{p1})15°$ anticlockwise and clockwise, and then we get two cutting paths $\boldsymbol{V}_{d1} = (C_{p0} - C_{p1})R(\boldsymbol{V}_{nf}, 15°)$, $\boldsymbol{V}_{d2} = (C_{p0} - C_{p1})R(\boldsymbol{V}_{nf}, -15°)$, where $R(\boldsymbol{V}_{nf}, 15°)$ is a rotation function.
(3) Taking $\boldsymbol{V}_{d1}$,$\boldsymbol{V}_{d2}$ as paths, and $\boldsymbol{V}_{n1} = \boldsymbol{V}_{nc} \times \boldsymbol{V}_{d1}$, $\boldsymbol{V}_{n1} = \boldsymbol{V}_{nc} \times \boldsymbol{V}_{d2}$ as normals, we can construct two planes P_{ln1}, P_{ln2} respectively.
(4) Cut S_v by P_{ln1}, P_{ln2}, we can get a series of intersection points $\{I_{pi}\}$, start from C_{p1}, according to the linked topology information of mesh edges, $\{I_{pi}\}$ can be sorted in sequence, and it is easy to intercept a part with a given length to form a C_L in the arrow shape.

N_L can be generated by several steps as follows, as shown in Fig. 17.

(1) According to the input point P, we acquire the nearest C_{Li} and line segment P_iP_{i+1}, and also get face C_{fi} that P_iP_{i+1} locates. Then map P to P_iP_{i+1} to get the endpoint P_{t0} of N_L.
(2) Generate the 3D shape of N_L similarly to the G_L generation method.
(3) According to the length proportion f_L of P_{t0} in its located boundary, find the corresponding 2D endpoint P_{t0_2d}, and then form the 2D shape of N_L.

N_L also can be generated from 2D. Based on f_L, N_L can be moved on the boundary of 3D and 2D patterns correspondingly, as N_{L0}, N_{L0_2d}, N_{L1}, N_{L1_2d} shown in Fig. 12.

Several feature style curves that pass through feature positions are frequently used in pattern design, such as neck line L_1, front and back center lines L_2, L_9, left and right side lines L_3, L_6, left and right armhole lines L_4, L_7, left and right shoulder lines L_5, L_8, as shown in Fig. 18(c). Utilizing the feature points, they can be generated automatically by the following steps:

(1) Recognize more feature points near armhole and on the bust, waist, hip, hem cross sections using the method listed in Section 3.1, as shown in Fig. 18(a). Since the feature points are located on the body surface, we can easy obtain the triangular *ID* T_{i_b} that a feature point F_{pi} locates.
(2) Map F_{pi} to S_v along the normal V_{ni_b} of T_{i_b} to get the projected point and face.
(3) From the recognized feature points, we can select the front center of neck, bust, waist, hip, hem and take their projected points and faces as input of $\{S_{pi}\}\{S_{fi}\}$ to generate the front center feature style curve L_2 according to the previous method, as shown in Fig. 18(b). Similarly, other feature style curves can also be generated by our purposeful selection of the projected points and faces as inputs of $\{C_{pi}\}$, $\{C_{fi}\}$ for C_L or $\{S_{pi}\}\{S_{fi}\}$ for F_L.

Feature style curves are helpful to reduce the interactive works in pattern design, as shown in Fig. 18(c), only L_{10} or L_{11} need be added for composing a common pattern.

5.3. Constraints of style curves in drawing

Each style curve has an actual meaning. Therefore constraints have to be introduced during free sketching to meet the requirements of crafting in pattern design.

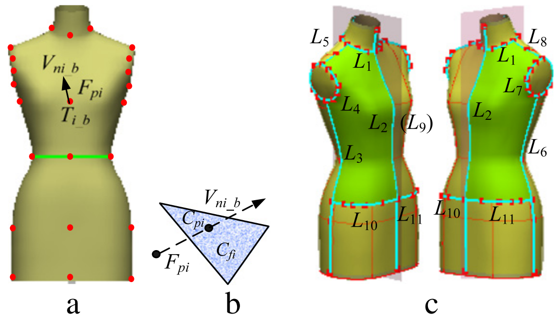
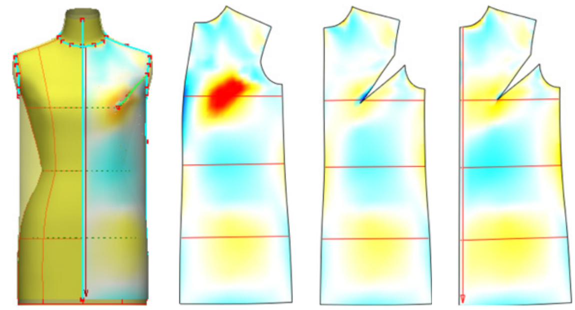
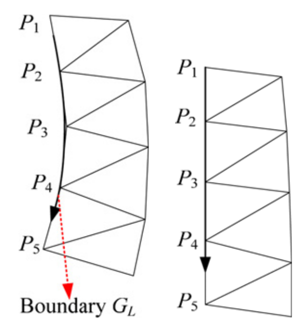


Fig. 18. Generation of some feature style curves.



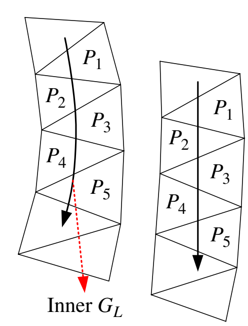
(a) Gauss curvature distribution. (b) 2D pattern flattened without D_L. (c) 2D pattern flattened with D_L. (d) 2D pattern flattened with D_L and G_L.

Fig. 19. Effects of D_L and G_L on pattern flattening.



(a) Before expanding. (b) After expanding.

Fig. 20. Start triangle strip flattening with boundary G_L.



(a) Before expanding. (b) After expanding.

Fig. 21. Start triangle strip flattening with inner G_L.

In this paper, we proposed several rules for drawing style curves.

(1) D_L should not pass through the whole region of a pattern; otherwise it will cut the pattern into two parts.
(2) G_L should not have sharp angles or turns, since the lengthwise direction of the pattern is almost straight.
(3) N_L should be on the pattern boundaries.

Based on these basic rules, constraints of style curve drawing are listed as follows.

(1) C_L and F_L should pass through no more than one endpoint of the same D_L, and also should not intersect with D_L.
(2) D_L should have no more than one endpoint on C_L or F_L, and cannot intersect with another D_L.
(3) G_L should be drawn in a 3D pattern region and each pattern has only one G_L.
(4) N_L should be drawn on the pattern boundaries. If a boundary is shared by two patterns, N_L should be generated in pairs for each pattern. If N_L is marked on a corner of a pattern, it should be generated in the angle bisector direction, and if the corner is shared with several patterns, N_L also should be generated for each pattern, as shown in Fig. 12.

Each style curve is at first generated and then checked against the previous constraints, and a style curve will be erased automatically if it does not satisfy these constraints, hence, the users only feel that it cannot be generated while the mouse is clicked.

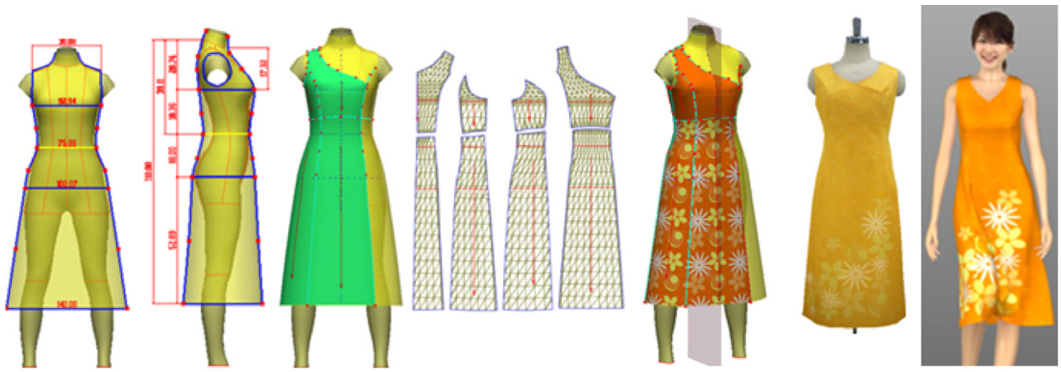
5.4. *The effect of style curves on pattern flattening*

How to enhance the quality of pattern flattening is an important issue. For dart lines, it is important for pattern makers to draw them as almost passing through the high curvature regions with smooth and freely editable shape. Here we take the following steps to make the dart pass through the high curvature parts by freely interactive sketching.

(1) Show the 3D surface curvature and 2D pattern deformation by color graph. The deformation of 2D pattern is calculated by the method presented by Sander [33] and Sheffer [34].
(2) Add darts as drawing C_L or F_L on the part with high curvature on 3D or high deformation on 2D, as shown in Fig. 19(a), (b).
(3) If we do not add darts in some cases, we can design the pattern to be smaller to make the pattern more developable, and then the deformation can be reduced, as shown in Fig. 24(b), (c).

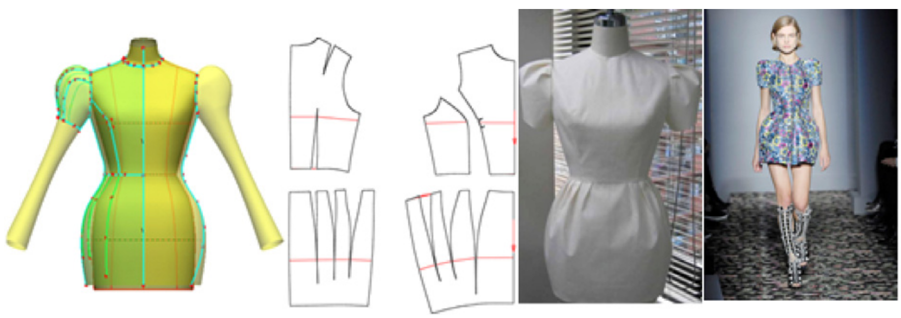
The direction of pattern boundaries should be along the lengthwise direction of the pattern as far as possible. However, few studies focus on this problem. In this paper, we add G_L to indicate the lengthwise direction, and adjust the process of pattern flattening to make the specified pattern boundary almost parallel with the lengthwise direction.

(1) Add G_L along with the lengthwise direction in a pattern, as shown in Fig. 19(a)
(2) Flatten the triangulated pattern using a mass–spring model, and the process of planar development is made faster by using triangle strips [35]. So, if there is a G_L in a pattern, we take some steps listed as follows for adjusting when flattening the first triangle strip.
(a) If a G_L is drawn on the boundary of a pattern, we take the boundary triangle strip as start triangle strip, and adjust it to be straight after being flattened, as shown in Fig. 20(a), (b).
(b) If a G_L is drawn on the inner part of a pattern, we take the triangles that G_L passes through as start triangle strip, and make the intersection points of G_L and triangles to be straight after being flattened, as shown in Fig. 21(a), (b).



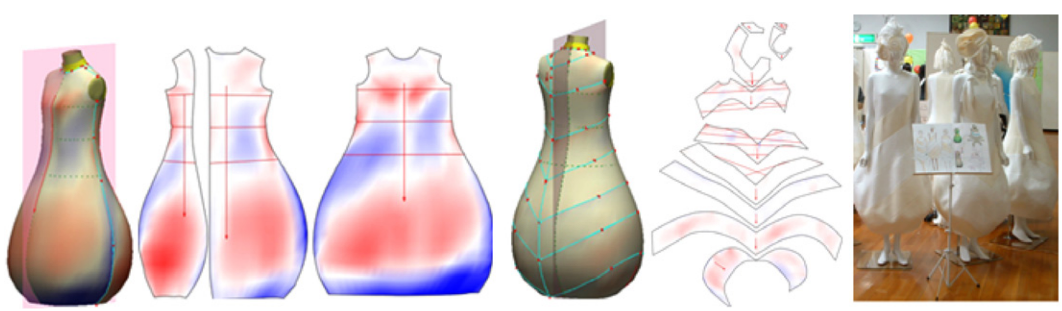
a b c d e

Fig. 22. Design examples of long dresses.



a b c d

Fig. 23. Design examples of short sleeve dresses.



a b c d e

Fig. 24. Design examples of artistic one-piece dresses.

Fig. 25. Design examples of cartoon dresses and trousers.

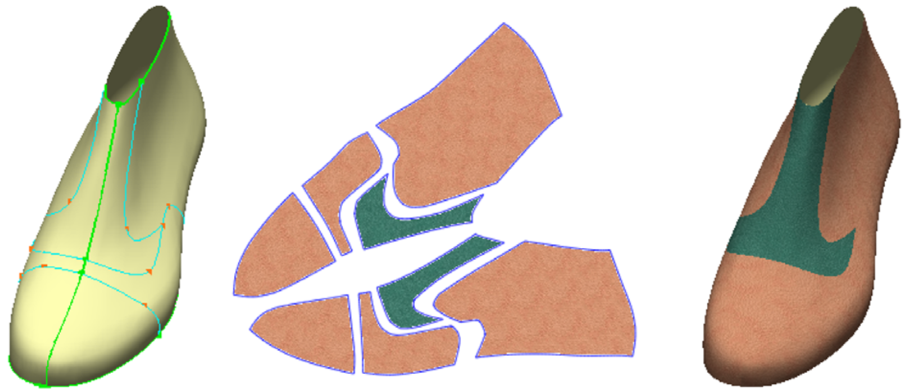


Fig. 26. Application of style curves in shoe pattern design.

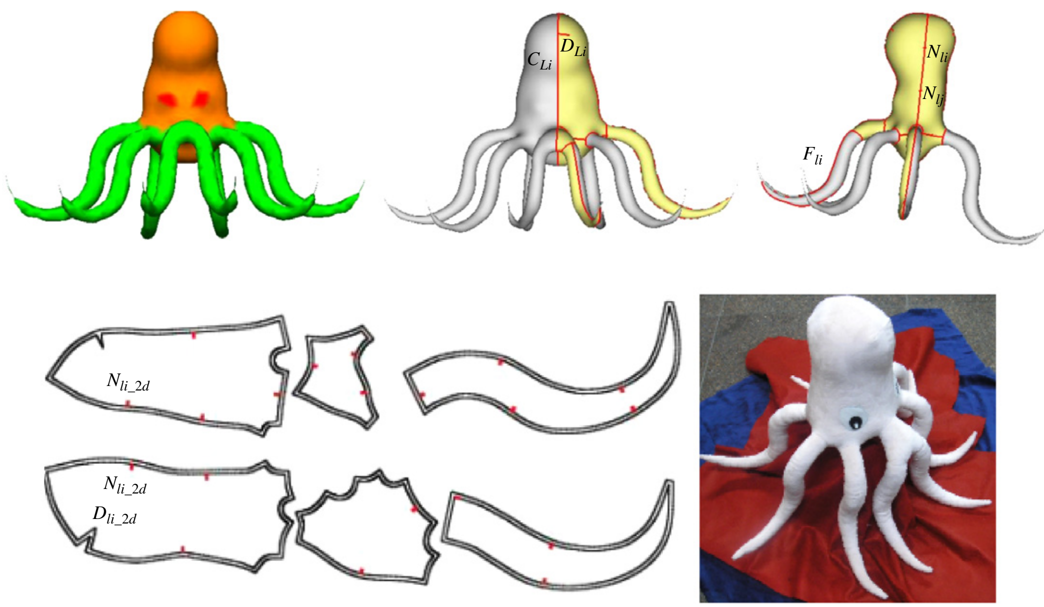


Fig. 27. Application of style curves in toy pattern design.

Note that, a G_L should be straight on 2D even if it is not straight on 3D. Fig. 19(c), (d) shows the pattern flattened without and with G_L respectively.

6. Examples of 3D garment interactive design through multiple parametric curves

Our works have been applied into the 3D garment design system, LookStailorX [36]. The following examples are all designed with this software. As shown in Fig. 22(a), the basic shape of the dress surface is edited by contour curves, and then 3D patterns can be quickly designed by interactively drawing style curves on the triangular mesh of the skirt, as shown in Fig. 22(b). 2D patterns are automatically flattened from 3D patterns, as shown in Fig. 22(c). Fig. 22(d) shows the flattened 3D/2D pattern with texture mapping, Fig. 22(e) shows the dressing result of the real dress sewn with 2D patterns. The right pattern on the upper part of one-piece dress is mirrored from the left one. Fig. 23(a) shows the surface and patterns of short sleeve dress designed by contour curves and style curves. Fig. 23(b), (c), (d) shows the 2D patterns and the dressing result of the sewn short sleeve dress. Fig. 24(a), (b) shows that the pattern of artistic one-piece dress has a larger distortion than that designed in a usual way. Fig. 24(c), (d) shows that the distortion is reduced by dividing the patterns into small ones. Fig. 24(e) shows the dressing result of the dress sewn with these small patterns. Fig. 25 shows the cartoon skirt and trousers designed by our method.

Our style curves also can be applied in other soft product design, such as shoes, toys and so on. Fig. 26 shows the design results of shoe patterns using our style curves C_L and F_L. Fig. 27 shows an example of complex toy patterns designed via our style curves C_L, F_L and N_L.

7. Conclusion and discussion

This paper presents a detailed and effective approach to solve the 3D garment interactive design problem based on constrained contour curves and style curves. It makes the 3D garment design more flexible, effective, valid and reasonable, which is important for developing commercial software in the apparel industry. Furthermore, 3D garment designs are clearly separated into surface modeling and pattern tailoring, and thus one surface style can match with different pattern styles, and vice versa. Such advantages are important for garment product family design. It is also meaningful to the soft product CAD systems in surface modeling and pattern designing, such as shoes, toys, furniture and so on. Especially, the style curves can be directly used to design patterns on arbitrary 3D triangular surfaces.

Our studies focus on garment design in a geometrical way. Next, we need to enhance the geometrical modeling ability for garment details and accessories, such as folds, tucks, pleats, pockets, buttons and strips. Additionally, based on the garment geometric modeling, simulation of realistic garment is also a research issue.

Acknowledgements

We would like to thank the National Natural Science Foundation of China (No. 60473129), the Ph.D. Programs Foundation of the Ministry of Education of China (No. 20060335118) for financially supporting this work, and the National High Technology Research and Development Program (863) of China (No. 2008AA6474711). We also would like to thank the fashion designers of Digital Fashion Ltd. and their partners, Mr. Kambara, and Miss Igarashi Katsuyo, who is a teacher of Kobe Fashion Art College, for designing artistic garment examples via our 3D design system.

References

[1] Gerber. http://www.gerber-technology.com.
[2] Lectra. http://www.lectra.com.
[3] Pad. http://www.padsystem.com.
[4] Durupynar F, Gudukbay U. A virtual garment design and simulation system. In: Proceeding of 11th international conference information visualization. 2007. p. 862–70.
[5] Metaaphanon N, Kanongchaiyos P. Real-time cloth simulation for garment CAD. In: Proceeding of ACM SIGGRAPH. 2005. p. 83–9.
[6] Bridson R, Marino S, Fedkiw R. Simulation of clothing with folds and wrinkles. In: Eurographics/SIGGRAPH symposium on computer animation. 2003 p. 28–36.
[7] Choi KJ, Ko HS. Stable but responsive cloth. In: Proceedings of SIGGRAPH. 2002. p. 604–11.
[8] Zhang DL, Yuen MMF. Cloth simulation using multi-level meshes. Computer Graphics 2001;25(3):383–9.
[9] Luo ZG, Yuan MMF. Reactive 2D/3D garment pattern design modification. Computer Aided Design 2005;37(6):623–30.
[10] Volino P, Courchesne M, Thalmann NM. Versatile and efficient technique for simulating cloth and other deformation objects. In: Proceedings of SIGGRAPH 95. NY (USA): ACM Press; 1995. p. 137–44.
[11] Volino P, Cordier F, Thalmann NM. From early virtual garment simulation to interactive fashion design. Computer Aided Design 2004;37:593–608.
[12] Turquin E, Cani MP, Hughes JF. Sketching garments for virtual characters. In: EG workshop on sketch-based interfaces and modeling. NY (USA): ACM Press; 2004. p. 175–82.
[13] Turquin E, Wither J, Boissieux L, Cani MP, Hughes JF. A sketch-based interface for clothing virtual characters. IEEE Computer Graphics & Application 2007; 27(1):72–81.
[14] Decaudin P, Julius D, Wither J. Virtual garments: A fully geometric approach for clothing design. Computer Graphics Forum 2006;25(3):625–34.
[15] Hinds BK, Mc Cartney J. Interactive garment design. Visual Computer 1990; 6(6):53–61.
[16] Mc Cartney J, Hinds BK, Seow BL. Dedicated 3D CAD for garment modeling. Journal of Materials Processing Technology 2000;107(11):31–6.
[17] Kim S, Kang T. Garment pattern generation from body scan data. Computer Aided Design 2002;35:611–8.
[18] Kim S, Park CK. Basic garment pattern generation using geometric modeling method. International Journal of Clothing Science and Technology 2007;19(1): 7–17.
[19] Wang CCL, Wang Y, Yuen MMF. Feature based 3D garment design through 2D sketches. Computer Aided Design 2002;35:659–72.
[20] Wang CCL, Wang Y, Yuen MMF. From laser-scanned data to feature human model: A system based on fuzzy logic concept. Computer Aided Design 2003; 35:241–53.
[21] Wang CCL, Tang K. Developability-preserved free-form deformation of assembled patches. In: Proceedings of the ninth ACM symposium on Solid modeling and applications. 2004. p. 231–6.
[22] Wang CCL, Wang Y, Yuen MMF. Design automation for customized apparel products. Computer Aided Design 2005;37(7):675–91.
[23] McCartney J, Hinds BK, Seow BL. The flattening of triangulated surfaces incorporating darts and gussets. Computer Aided Design 1999;31(4):249–60.
[24] Wang CCL, Smith S, Yuen MMF. Surface flattening based on energy model. Computer Aided Design 2002;34(11):823–33.
[25] Wang CCL. Freeform surface flattening based on fitting a woven mesh model. Computer Aided Design 2005;27(8):799–814.
[26] Wang CCL, Wang Y, Tang K. Reduce the stretch in surface flattening by finding cutting paths to the surface boundary. Computer Aided Design 2004;36(8): 665–77.
[27] Chen L, Lu GD, Wang J. Parametric 3D sleeve modeling based on hybrid dimension. In: International technology and innovation conference. 2006. p. 944–50.
[28] Zhang JW. C-curves: An extension of cubic curves. Computer Aided Geometric Design 1996;12(3):199–217.
[29] Zhang JW. Two different forms of C-B-Splines. Computer Aided Geometric Design 1997;14(1):31–41.
[30] Zhang JW. C-Bezier curves and surfaces. Graphical Model and Image Processing 1999;61(1):2–15.
[31] Wang J, Lu GD, Chen L. 3D garment surface design using sketch- based modeling method. In: International technology and innovation conference. 2006. p. 726–32.
[32] SGI. OpenGL software development kit. http://www.opengl.org/sdk/docs/man/.
[33] Sander PV, Snyder J, Ggortler SJ, Hoppe H. Texture mapping progressive meshes. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM Press; 2001. p. 409–16.
[34] Sheffer A, Levy B, Mogilnitsky M, Bogomyakov A. ABF++: Fast and robust angle based flattening. ACM Transactions on Graphics 2005;24(2):311–30.
[35] Li JT, Zhang DL, Lu GD. Flattening triangulated surface using a mass-spring model. International Journal of Advanced Manufacturing Technology 2005; 25(1–2):108–17.
[36] DressingSim. http://www.dressingsim.com.

Jin Wang is currently a Post Doctor at the college of Mechanical and Energy Engineering, Zhejiang University, China. He gained his B.Sc. and Ph.D. in Mechanical Engineering from Zhejiang University, China, in 2003 and 2008. His research interests include CAD/CAM, design automation and optimization, geometric modeling, reverse engineering and computer graphics.

Guodong Lu is currently a Professor at the College of Mechanical and Energy Engineering, Zhejiang University, China. He is the Deputy Director of Engineering and Computer Graphics Institute at Zhejiang University. He received his B.Sc., M.Sc. and Ph.D. from Zhejiang University, China, in 1983, 1990 and 2000. His research interests include CAD/CAM, computer graphics, 3D reconstruction and design methodology.

Weilong Li is currently a doctor candidate at Zhejiang University, China. He received his M.Sc. in Mechanical Engineering from Guangxi University in 2003 and B.Sc. from Central South University in 1996. His research interests include CAD/CAM, computer graphics and imaging, modeling and simulation of garment, and design methodology.

Long Chen is currently a lecturer at the college of Mechanical Engineering, University of Shanghai for Science and Technology, China. He gained his B. Eng. from College of Logistics Engineering, Wuhan University of Technology, China, in 2002 and Ph.D. in Mechanical Engineering from Zhejiang University, China, in 2008. His research interests include CAD/CAM/ERP, reverse engineering, engineering optimization, computer graphics.

Yoshiyuki Sakaguti is the President of Digital Fashion Ltd. He received his Ph.D. in 1997 from Kyoto University, Japan. He has researched in 3D garment design for more than 15 years. His interests include CAD/CAM, Computer Graphics, especially on the realization of Magic Mirror and construction of Fashion On Demand.