



## Pattern design on 3D triangular garment surfaces<sup>\*</sup>

WANG Jin<sup>†1</sup>, LU Guo-dong<sup>†‡1</sup>, LI Ji-tuo<sup>2</sup>, CHEN Long<sup>1</sup>, ZHANG Dong-liang<sup>3</sup>

<sup>(1)</sup>State Key Lab. of CAD & CG, Zhejiang University, Hangzhou 310027, China)

<sup>(2)</sup>Institute of Automation, Chinese Academy of Sciences, Beijing 100080, China)

<sup>(3)</sup>Digital Fashion Ltd., Osaka 5410053, Japan)

<sup>†</sup>E-mail: dwjcom@cmee.zju.edu.cn; lugd@cmee.zju.edu.cn

Received Jan. 9, 2007; revision accepted Apr. 12, 2007

**Abstract:** This paper focuses on a pattern design method for a 3D triangular garment surface. Firstly, some definitions of 3D style lines are proposed for designing the boundaries of patterns as drawing straight lines or splines on the triangular surface. Additionally some commonly used style lines are automatically generated to enhance design efficiency. Secondly, after style lines are preprocessed, a searching method is presented for quickly obtaining the boundaries and patches of a pattern on the 3D triangular surface. Finally a new pattern design reuse method is introduced by encoding/decoding the style line information. After style lines are encoded, the pattern design information can be saved in a pattern template and when decoding this template on a new garment surface, it automates the pattern generation for made-to-measure apparel products.

**Key words:** Pattern design, Style lines, 3D patch searching, Encoding/decoding, Design reuse

**doi:**10.1631/jzus.2007.A1642

**Document code:** A

**CLC number:** TP391.72

### INTRODUCTION

Garment design presently is mostly based on 2D CAD technique, which gradually turns to bottleneck in the apparel industry. 2D grading techniques in commercial apparel CAD systems (www.lectra.com; www.gerberotechnology.com) are mostly implemented by grading the pattern boundary curves. Such solution is not 3D intuition, and is hard to make the garment fit to the body.

Cloth simulation techniques (Volino *et al.*, 1995; Zhang and Yuen, 2001; Bridson *et al.*, 2003; Liu *et al.*, 2003; www.padsystem.com) provide a way for testing patterns by assembling 2D patterns in a computer system and draping them on a virtual human body. However, apparel simulation cannot provide valid methods for editing the 3D apparel surface to quickly make the apparel fit. Recently, an online made-to-

measure system was presented by Cordier *et al.* (2003). Their system allows interactive adjustment of the 3D mannequin according to the shopper's body measurements, online resizing of the garment to fit the mannequin, and real-time cloth simulation of the garment corresponding to body motion. However, their approach is also based on 2D pattern design and draping simulation, and their major purpose is for visualization and animation, not for design.

Currently, more and more studies focus on designing a 3D garment directly on a 3D scanned body and then generating 2D patterns via flattening 3D patterns. In Hinds and McCartney's work (Hinds and McCartney, 1990; McCartney *et al.*, 2000), the garment is represented as a collection of panels offsetting from the body surface, and the garment panels are designed around the static mannequin body. Based on the human features, Wang *et al.* (2003a; 2003b) generated a 3D garment by piecewise construction and mesh subdivision technique. However, these studies focused on constructing the garment surface, and could only generate simple 3D patterns. Turquin *et al.* (2004) and Decaudin *et al.* (2006) drew garment

<sup>‡</sup> Corresponding author

<sup>\*</sup> Project supported by the National Natural Science Foundation of China (No. 60473129) and the Ph.D Programs Foundation of the Ministry of Education of China (No. 20060335118)

contours and seamlines of the garment directly on a virtual mannequin at the front and back views, and then converted the sketch into an initial 3D surface using an existing method based on a precomputed distance field around the mannequin. The method designs the 3D garment patterns during construction of a 3D garment surface. However, a 2D input point is hard to acquire its precise 3D coordinate if it is not mapped onto a referenced 3D surface, so their method can only design the patterns from the front and back views, with its design process being a 2D method.

Actually, the process of 3D garment design includes 3D garment surface design and 3D garment pattern design. Hinds, McCartney and Wang only focused on the surface construction, while Turquin *et al.*(2004) and Decaudin *et al.*(2006) designed the patterns combined with constructing a garment surface which is hard to achieve freely in 3D garment design.

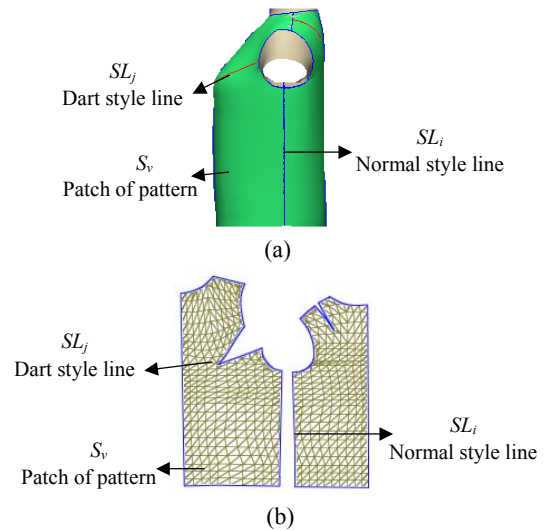
In this paper, we provide a solution for designing 3D patterns on a 3D garment surface generated by an interactive sketch technique (Li *et al.*, 2005) on 3D scanned bodies. Our main contributions are listed as follows:

- (1) Providing some proper definition and drawing methods for interactive design or automatically generating pattern style boundaries on a 3D triangular garment surface;
- (2) Providing an intelligent method for searching the boundaries and patches of a pattern on a 3D triangular surface, with several preprocessings of the style lines related with apparel design being handled;
- (3) Providing an encoding/decoding method of the style lines on a triangular surface for garment design reuse or 3D pattern grading.

**BOUNDARIES DESIGN OF PATTERN ON 3D TRIANGULAR SURFACE**

A triangular surface  $S_v=(P, E, T)$  is represented by a set of vertices  $P=\{P_i\}$ , triangles  $T=\{T_j\}$  and edges  $E=\{E_k\}$ . The topology linkage information is also defined:  $P_i=\{\{AE_i\}, \{AT_i\}\}$ ,  $T_j=\{(P_{j0}, P_{j1}, P_{j2}), (E_{j0}, E_{j1}, E_{j2}), \{AT_j\}\}$ ,  $E_k=\{(P_{k0}, P_{k1}), \{AE_k\}, \{AT_k\}\}$ . 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.

Define a 3D Pattern  $Ptn=(\{SL_i\}, S_v)$ .  $\{SL_i\}$  is a series of style lines composed of the boundaries of a pattern and  $S_v$  denotes the triangular patch of a pattern, as shown in Fig.1.



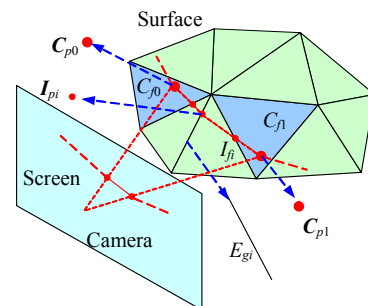
**Fig.1 Design of 3D patterns (a) and flatten 2D patterns (b)**

**3D pattern boundaries design by drawing straight style lines interactively**

Define a style line generated by drawing straight lines on  $S_v$  be  $CL=(C_p, C_f, E_g, I_p, I_f, ID, IDA)$ , where the meanings of  $C_p, C_f, E_g, I_p, I_f, ID, \{IDA_i\}$  are listed in Table 1 and shown in Fig.2.

**Table 1 Parameters used to represent CL style lines**

Parameters	Meanings
$C_p=\{C_{p0}, C_{p1}\}$	The endpoints of CL
$C_f=\{C_{f0}, C_{f1}\}$	ID of the triangles where $C_p$ is located
$E_g=\{E_{gi}\}$	Edges sets that CL goes through
$I_p=\{I_{pi}\}$	Intersection points of CL and $E_g$
$I_f=\{I_{fi}\}$	ID of the triangles that $I_p$ locates
ID	ID of CL
$IDA=\{IDA_i\}$	ID sets of the style lines linking with CL



**Fig.2 Definition of CL style lines**

Two control points and triangles are needed to create a *CL* style line, and they can be acquired from interactively drawing process on  $S_v$  by the following steps:

(1) Utilize the method mentioned in (Wang et al., 2004) to pick out the triangles ID  $f_i$  after a 2D point  $p(x, y)$  input on the initialized OpenGL scene.

(2) Convert the 2D point  $p=(x, y)$  to a 3D point  $vt=(x, y, z)$  in the OpenGL scene. Let the OpenGL model-view matrix be  $M$ , projection matrix be  $P$ . Here  $M$  and  $P$  are  $4 \times 4$  matrix. Let  $W=MP$ ,  $V=(0, 0, 0, s)$ , we can convert  $p$  to  $vt$  by the following equations:

$$V=VW, \tag{1}$$

$$V_x=(p_x-w/2)V_w(w/2)^{-1}, \tag{2}$$

$$V_y=(p_y-h/2)V_w(h/2)^{-1}, \tag{3}$$

$$V=VW^{-1}, \tag{4}$$

$$vt=(V_x, V_y, V_z), \tag{5}$$

where  $s$  is a scale coefficient described by OpenGL, and its default value is 1.0.  $V_x, V_y, V_z, V_w$  are the  $x, y, z, w$  coordinate of  $V$ , respectively;  $p_x, p_y$  are the  $x, y$  coordinate of  $p$ , respectively;  $w$  and  $h$  are respectively the width and height of the OpenGL scene region.  $(p_x-w/2)$  and  $(p_y-h/2)$  mean that the original OpenGL point of the world coordinate system is on the center of the rendering scene.

(3) Project  $vt$  onto  $S_v$ . Convert 2D point  $p$  to two 3D points  $vt_1, vt_2$  by taking two different depth points  $v_{4,1}=(0, 0, 0, w)$  and  $v_{4,2}=(0, 0, 1, w)$  in local system as reference, and then we can get a project direction  $vt_{Dir}=vt_2-vt_1$  along the  $z$ -axis direction of the global coordinate system. The projection point on  $S_v$  of  $vt$  can be regarded as the intersection point  $pt$  of triangle  $f_i$  and  $vt_{Dir}$ .

(4) Get the projection triangle ID  $C_{f1}, C_{f2}$  and project points  $C_{p1}$  and  $C_{p2}$  of  $CL$  by twice mouse clicking on the surface as input.

Taking the average normal of  $C_{f1}, C_{f2}$  as the projection direction, the information of  $E_g, I_p, I_f$  that  $CL$  goes through is obtained and thus a  $CL$  is generated.

### 3D pattern boundaries Design by drawing spline style lines interactively

To design the boundaries of a pattern more freely and create more complex shapes, we generate them by drawing splines on  $S_v$ , and call them *FSPL* style lines, as shown in Fig.3.

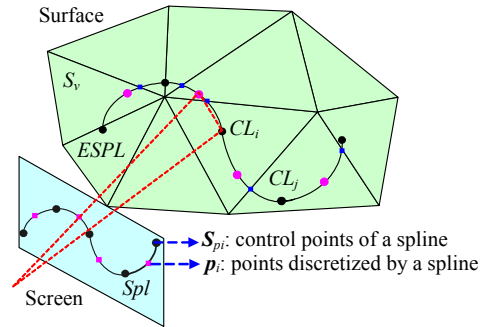


Fig.3 Definition of *FSPL* style line

$FSPL=(S_{pl}, \{S_{fi}\}, \{CL_i\}, ID, IDA)$ , where  $S_{pl}$  is a cubic spline containing a series of control points  $\{S_{pi}\}$ .  $\{S_{fi}\}$  are the IDs of the triangles where  $\{S_{pi}\}$  are located.  $\{CL_i\}$  is the *CL* sets which are made up of *FSPL*. An *FSPL* can be generated by the following steps:

(1) Obtain a series of 3D points  $\{P_i\}$  discretized by  $S_{pl}$  as the input points for  $\{CL_i\}$ .

(2) Map  $\{P_i\}$  onto  $S_v$  to get a series of points and triangles IDs of these points located as  $\{C_{pi}\}$  and  $\{C_{fi}\}$  of  $\{CL_i\}$ .

(3) Generate a series of *CL* as presented in Section 2.1 and convert them into  $\{CL_i\}$  of *FSPL*.

Note that  $\{CL_i\}$  is the actual element for an *FSPL* that is handled in obtaining, trimming and flattening a pattern patch.

### Generating feature style lines automatically

Some style lines that pass through feature positions of the garment are used frequently in pattern design, and these style lines are called feature style lines, as shown in Fig.4. We generate them as draw splines for freely designing or modification on  $S_v$ . Note that, only  $\{S_{pi}\}, \{S_{fi}\}$  are needed for generating an *FSPL* style line. So they can be automatically generated by the following steps:

(1) Utilize the method presented in (Yan et al., 2005) to recognize some necessary feature points and obtain the IDs of the triangles where they are located.

(2) Generate some *FSPL* by taking the recognized feature points and triangles' IDs as input for  $\{S_{pi}\}, \{S_{fi}\}$  of *FSPL*.

### Preprocessing of style lines

The following preprocessing of style lines has to be handled for obtaining the boundaries and patches of a pattern on 3D garment surface.

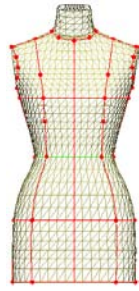


Fig.4 Feature style lines

(1) Break the intersection style lines and record their linkage information. If a style line intersects with other style lines, such as  $L_1$  or  $L_2$  shown in Fig.5a, it should be broken into two new style lines on the intersection,  $L_1, L_2$  or  $L_3, L_4$  as shown in Fig.5b, and the linkage information of the new style lines must be recorded. The new style lines are actually elements for obtaining and trimming a pattern.

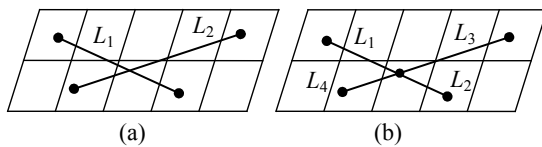
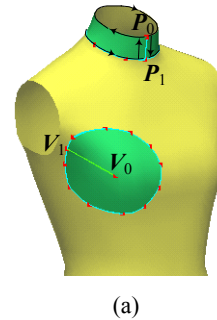


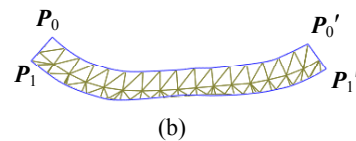
Fig.5 Breaking style lines at the intersection points  
(a) Before breaking; (b) After breaking

(2) Label the bridge style line. A bridge style line is used to merge two closed style line loops into one, as the style line  $P_0P_1$  shown in Fig.6. According to the definition of a bridge style line, it will be searched twice in the process of searching a closed style line loop. So we can regard it as two parallel style lines with identical position when creating the patch boundaries, trimming and flattening a patch, as  $P_0P_1$  and  $P_0'P_1'$  shown in Fig.6b. However, it is not used for checking whether a vertex is inside or outside during the process of labelling inside vertices of a pattern.

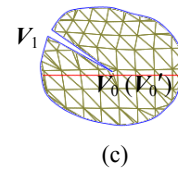
(3) Label the dart style line. Darts are triangular elements of a pattern that enables a flat piece to take the shape of a curved surface once sewn. When a surface has a large deformation after being fitted to the 3D space, usually, appropriate darts should be inserted into the surface, as  $V_0V_1$  shown in Fig.6. A dart style line cannot be used to make up patch boundaries or be taken as reference for checking whether a vertex is inside or outside the patch. While in trimming and flattening a patch, it is regarded as



(a)



(b)



(c)

Fig.6 Pattern boundaries with bridge or dart style lines. (a) 3D pattern; (b) 2D pattern with bridge style lines; (c) 2D pattern with dart style lines

two parallel style lines with identical position, as  $V_0V_1$  and  $V_0'V_1'$  shown in Fig.6.

(4) Process the boundaries of  $S_v$ . If  $S_v$  has boundaries, we should utilize the information of vertices, edges and faces on the boundaries to generate some style lines.

(5) Adjust the style lines position. If a style line passes through a vertex of a triangle on  $S_v$ , we should move it into the triangle a little. So all the vertices of  $S_v$  are on different sides of a style line, and we can easily label whether a vertex is inside or outside a patch with its closed boundaries as reference.

## OBTAINING 3D PATTERNS FROM STYLE LINES

The process of obtaining a 3D pattern after a start point is input by one mouse click and can be listed as follows:

(1) Search a closed loop of style lines as pattern boundaries which have the nearest style line to the input point. Record the edges intersecting with the loop and the number of intersection points of each intersection edge.

(2) Label all inside vertices of the targeted pattern patch enclosed by the closed loop of style lines.

(3) Extract the triangles partially or wholly inside the pattern boundaries to construct a mesh of the pattern patch.

### Searching the boundaries for a 3D pattern

To obtain the closed style line loops as boundaries of a pattern patch, firstly we should utilize the following steps to obtain a style line on the loop, which is called a start style line,  $SL$  for short.

(1) Get the triangle  $T_0$  containing the input point  $P_0$ .

(2) If there are style lines passing through  $T_0$ ,  $SL$  is the nearest one to  $P_0$ .

(3) If there is no style line intersecting with  $T_0$ , get a vertex on  $T_0$  as the starting point. Utilize the breadth-first search method to search out an edge intersecting with some style lines in the linkage edges of  $T_0$ . Suppose the edge is  $E_i$ , and the corresponding vertex is  $P_j$ ,  $SL$  is the one whose intersection is the nearest to  $P_j$ .

Let a style line on the closed loop be  $BL$ , which is composed of boundaries of the pattern patch, and let the being processed style line be  $CurL$ . Starting from  $SL$ , the pattern patch boundaries can be searched as follows:

(1) Set a  $CurL$  to be a  $BL$ .

(2) Check whether a  $CurL$  links with other style lines. If “yes”, go to (3), otherwise go to (5).

(3) Calculate all the angles of  $CurL$  and its linked style lines, and sort its linked style lines by the angles in an increasing order, such as  $\{AL_1, AL_2, \dots, AL_i, \dots, AL_n\}$ .

(4) Set the linked style line  $AL_i$  to be  $CurL$  in turn. If  $CurL$  is identical to  $SL$ , it means a closed loop is obtained and can be returned with “true”; otherwise, go to (1) and repeats the process until  $CurL$  is identical to  $SL$ .

(5) Set  $CurL$  not to be a  $BL$  and return false.

As shown in Fig.7, when the input  $P_0$  is  $M$ , its corresponding  $SL$  is  $L_1$ , and the pattern patch boundaries are  $\{L_1, L_4, L_5, L_6\}$  after previous processing steps being implemented.

Note that, the bridge style line should be regarded as two parallel style lines with identical position and the dart style lines should be ignored in the process presented in this section.

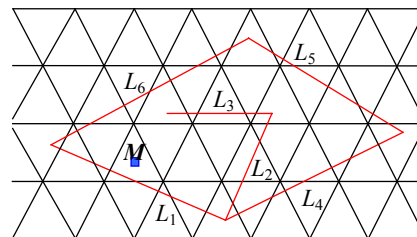


Fig.7 Searching the boundaries of a 3D pattern

### Label inside vertices of a pattern patch

To label the inside vertices of the pattern patch, firstly we should obtain a start inside vertex,  $SP$  for short.  $SP$  can be obtained by the following steps:

(1) Get the triangle  $T_0$  which contains the input point  $P_0$ .

(2) If there are no style lines passing through  $T_0$ , then  $SP$  is the vertex on  $T_0$  which is the nearest to  $P_0$ . As shown in Fig.8,  $P_1$  is  $SP$  corresponding to the input point  $M_1$ .

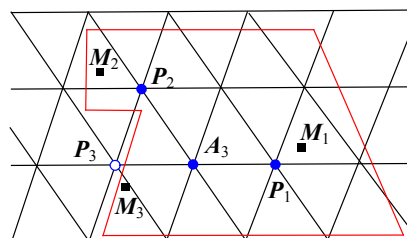


Fig.8 Obtaining a start inside the vertex

(3) If there are style lines passing through  $T_0$ , we try to find out all the vertices  $\{P_i\}$  in  $T_0$  that has no intersection between  $P_0P_i$  and the style line loop, and set the one nearest to  $P_0$  to be  $SP$ , such as  $P_2$  is  $SP$  corresponding to the input point  $M_2$ , as shown in Fig.8. If we cannot find such a vertex, it means all the vertices in  $T_0$  are outside vertices. We take a vertex  $P_i$  in triangle  $T_0$  which is the nearest to  $P_0$ , and suppose the linked vertices of  $P_i$  are  $\{A_i\}$ . Try to find a linked vertex of  $P_i$  that has an odd intersection number between  $A_iP_i$  and the style line loop, and set such vertex  $A_i$  as  $SP$ . As shown in Fig.8,  $P_3$  is an outside vertex and  $A_3$  is  $SP$  corresponding to  $M_3$ .

Starting from  $SP$ , we use the “breadth-first search” method to search out all the inside vertices of the targeted pattern patch. Let the inside vertex be  $IP$ , and its linked edges be  $\{E_i\}$ , another vertex on  $E_i$  is

$E_{iAP}$ . Similarly let the outside vertex be **OP**. Labelling process of inside vertices can be listed as follows:

Push  $SP$  into a stack, and do the following operations while the stack is not null.

(1) Pop out the vertex on the stack top and label it as **IP**.

(2) Search  $\{E_i\}$  of **IP**, if there is zero or even  $BL$  passing through  $E_i$  and  $E_{iAP}$  has not been labelled as **IP** or **OP**, push  $E_{iAP}$  into the stack. If there is odd  $BL$  passing through  $E_i$ , we label  $E_{iAP}$  as **OP**.

(3) If there are unsearched style lines passing through  $E_i$ , utilize the method as mentioned in Section 3.1 to check whether there is another closed loop inside the previous searched loop. If a new closed loop exists, label the style lines on the new loop as  $BL$  and go to (2) to continue the inside vertex labelling process.

Fig.9a shows the **IP** and **OP** searched out after first step of labelling. During labelling, a new loop  $V_1V_2V_3V_4$  is checked. Only the first layer loops inside  $P_1P_2P_3P_4$  need to be considered, such as  $V_1V_2V_3V_4$ . Other layer loops inside the first layer can be ignored, such as  $D_1D_2D_3$ , since they are outside the targeted pattern patch actually. Fig.9b shows all the inside vertices that are labelled after previous labelling steps, where the solid dot is **IP** and hollow dot is **OP**.

Note that, a style line has recorded all the edges ID that it goes through. So no special calculation is needed during the inside vertex labelling, high efficiency is achieved. The bridge style lines and dart style lines are ignored in the process presented in this subsection.

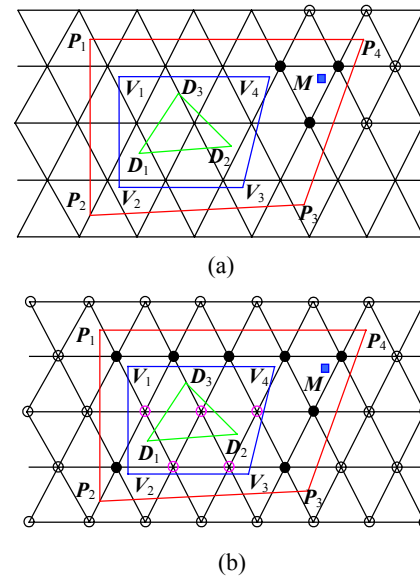
### Constructing the mesh of a pattern patch

The mesh of a pattern patch is composed of its boundaries and the triangles partially or wholly inside the patch on the original surface. Following are the steps to construct a new mesh of the pattern patch.

(1) Get the intersecting points of  $BL$  and triangles partially inside the patch to create one or several polygons. Vertices in the polygon should be stored in anti-clockwise order which is identical to that saved in the triangles of the original garment mesh.

(2) Utilize the polygons to triangulate the triangles partially inside the patch.

(3) Remove the triangles from the original apparel mesh and insert the newly triangulated triangles. Build the topologic relationship between the triangles



**Fig.9 Pattern patch inside vertices labelling process**  
(a) After first labelling; (b) When the labelling is finished

inside the target pattern patch and a final patch mesh is obtained.

(4) Flatten a mesh of 3D pattern patch using the method presented in (Li *et al.*, 2005), and a corresponding 2D mesh can be obtained as shown in Fig.10.

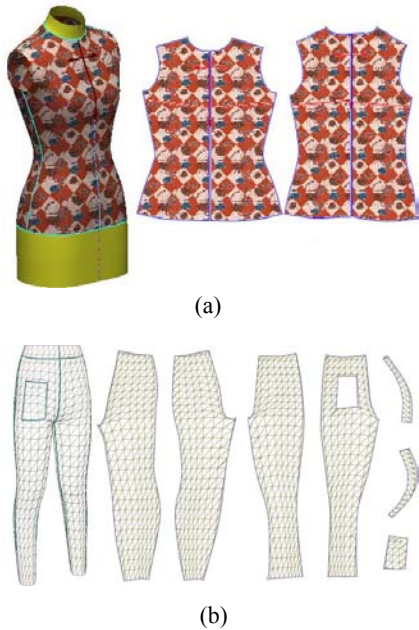
From Fig.10, we can learn that the feature style lines generated automatically can really enhance efficiency in 3D pattern design.

### 3D PATTERN DESIGN REUSE BASED ON ENCODING/DECODING OF STYLE LINES

The style line contains the pattern design information. After encoding the style lines, the designed pattern can be stored as a pattern style template by decoding the style lines, the pattern style can be mapped onto a new surface, and new pattern patches can be duplicated after searching the boundaries and inside vertices again. So the encoding/decoding process of style lines is essential for a pattern design reuse.

#### Encoding/decoding style lines based on barycentric mapping

Note that, the basic element of the style lines is  $CL=(C_p, C_f, E_g, I_p, I_f, ID, IDA)$ . The topologic relation



**Fig.10 3D and 2D pattern generation after trimming. (a) 3D and 2D jacket pattern with boundaries composed by feature style lines only; (b) 3D and 2D trousers pattern with boundaries composed by feature style lines and freely designed style lines**

of  $C_p, I_p$  and  $C_f, I_f$  is fixed if the topology of  $S_v$  keeps unchanged. So we can encode  $C_p, I_p$  with  $C_f, I_f$  through barycentric mapping. Let  $Q_i$  be a point of  $C_p$  or  $I_p$  and  $T_i$  its corresponding triangle.  $T_i = \{P_{i0}, P_{i1}, P_{i2}\}$ , and then  $Q_i$  can be encoded by the area ratio of triangle  $\{Q_i, P_{i1}, P_{i2}\}, \{Q_i, P_{i0}, P_{i1}\}, \{Q_i, P_{i0}, P_{i2}\}$  and  $T_i$ , as shown in Fig.11 and Eqs.(6)~(8).

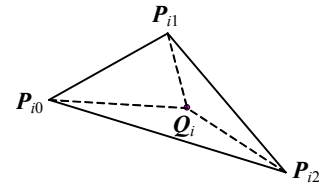
$$Q_i = f(T_i, u_0, v_0, w_0), \quad (6)$$

$$u_0 = \frac{u}{u+v+w}, v_0 = \frac{v}{u+v+w}, w_0 = \frac{w}{u+v+w}, \quad (7)$$

$$\begin{cases} u = \frac{\|Q_i P_{i1} \times P_{i2} P_{i1}\|}{\|P_{i0} P_{i1} \times P_{i2} P_{i1}\|}, \\ v = \frac{\|Q_i P_{i0} \times P_{i2} P_{i0}\|}{\|P_{i1} P_{i0} \times P_{i2} P_{i0}\|}, \\ w = \frac{\|Q_i P_{i0} \times P_{i1} P_{i0}\|}{\|P_{i2} P_{i0} \times P_{i1} P_{i0}\|}, \end{cases} \quad (8)$$

where  $u_0, v_0, w_0$  are the encoding of  $Q_i$  with  $T_i$  being taken as the reference, and its corresponding decoding is expressed as:

$$Q_i = u_0 P_{i0} + v_0 P_{i1} + w_0 P_{i2}. \quad (9)$$



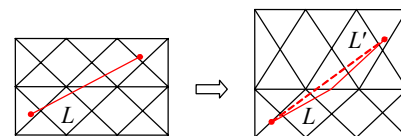
**Fig.11 Encoding/decoding via barycentric mapping**

If  $(Q_i P_{i1} \times P_{i2} P_{i1}) \cdot (P_{i0} P_{i1} \times P_{i2} P_{i1}) < 0$ , it means  $Q_i$  and  $P_{i0}$  are on the different sides of  $P_{i2} P_{i1}$ , and set  $u = -u$ . Similarly

if  $(Q_i P_{i0} \times P_{i2} P_{i0}) \cdot (P_{i1} P_{i0} \times P_{i2} P_{i0}) < 0$ , set  $v = -v$ ;  
if  $(Q_i P_{i0} \times P_{i1} P_{i0}) \cdot (P_{i2} P_{i0} \times P_{i1} P_{i0}) < 0$ , set  $w = -w$ .

**Total or partial encoding/decoding of style lines**

Total encoding/decoding of the style lines means encoding or decoding all the information that needs to be saved, such as  $C_p$  and  $I_p$  for a  $CL$  style line, and  $Spl, \{CL_i\}$  for an  $FSPL$  style line. This method can keep the topology correlations of style lines well during mapping of the template of style lines. However, it will bring on unexpected distortion if the garment surface has larger deformation. As shown in Fig.12a,  $L$  is an original  $CL$  style line, and it is distorted after being mapped onto a newly deformed surface as shown in Fig.12b.



**Fig.12 Shape distortion of the style lines after decoding totally. (a) Initial shape of L; (b) Shape of L after mapping**

In order to make the style lines smooth after template mapping, we adopt a partial encoding/decoding method, which encodes  $C_p$  only for a  $CL$  style line and generates it after  $C_p$  is decoded on the new surface. For an  $FSPL$  style line, it encodes  $Spl$  only and generates a new one by having its  $Spl$  decoded. This method can make  $CL$  nearly straight and  $FSPL$  nearly smooth after being decoded.

Fig.13 shows a pattern template mapping onto a tight garment surface and a designed garment surface.



**Fig.13 Pattern style template mapping on different surfaces. (a) Mapping on a jacket surface; (b) Mapping on a one-piece dress surface**

## CONCLUSION AND DISCUSSION

Our paper presents a basic solution for pattern design on triangular surface including generating the style line, obtaining 3D pattern patch, and mapping pattern template.

The boundaries of a 3D pattern patch are designed as drawing straight lines or splines on the triangular mesh, and complex shape of the pattern patch can be freely obtained. The 3D pattern patch can be conveniently obtained with one mouse click as the input after some preprocessing of style lines. The style of the pattern patch is easily saved and mapped onto a new garment surface by using the totally or partially encoding/decoding method of style lines.

Additionally, some commonly used feature style lines are automatically generated by recognizing the human feature points. These feature style lines can compose the boundaries of 3D pattern patch directly or be taken as the reference for pattern design.

The style drawing mode, 3D pattern patch obtaining method and pattern template mapping method can also be applied to any 3D triangular design process, such as shoe design and toy design. However, there are still some work left in pattern template mapping, e.g., to keep the topologic correlations between style lines unchanged after they are partially encoded/decoded, and to satisfy the needs that the topology of the triangular surface can be changeable during mapping of the style lines template.

## References

- Bridson, R., Marino, S., Fedkiw, R., 2003. Simulation of Clothing with Folds and Wrinkles. *EUROGRAPHICS /SIGGRAPH Symp. on Computer Animation*, p.28-36.
- Cordier, F., Seo, H., Thalmann, N.M., 2003. Made-to-measure technologies for an online clothing store. *IEEE Computer Graph. Appl.*, **23**(1):38-48. [doi:10.1109/MCG.2003.1159 612]
- Decaudin, P., Julius, D., Wither, J., Boissieux, L., Sheffer, A., Cani, M.P., 2006. Virtual garments: a fully geometric approach for clothing design. *Computer Graphics Forum*, **25**(3):625-634. [doi:10.1111/j.1467-8659.2006.00982.x]
- Hinds, B.K., McCartney, J., 1990. Interactive garment design. *The Visual Computer*, **6**(2):53-61. [doi:10.1007/BF0190 1066]
- Li, J.T., Lu, G.D., Zhang, D.L., 2005. Individuality based garment surface modeling through a 3D interactive sketch. *J. Computer-Aided Design & Computer Graphics*, **17**(11): 2512-2517 (in Chinese).
- Liu, H., Chen, C., Shi, B.L., 2003. Simulation of 3D garment based on improved spring-mass model. *J. Software*, **14**(3):619-627 (in Chinese).
- McCartney, J., Hinds, B.K., Seow, B.L., Gong, D., 2000. Dedicated 3D CAD for garment modeling. *J. Mater. Processing Technol.*, **107**:31-36. [doi:10.1016/S0924- 0136(00)00695-6]
- Turquin, E., Cani, M.P., Hughes, J.F., 2004. Sketching Garments for Virtual Characters. *EG Workshop on Sketch-based Interfaces and Modeling*, p.175-182. [doi:10.1145/ 1185657.1185781]
- Volino, P., Courchesne, M., Thalmann, N.M., 1995. Versatile and Efficient Technique for Simulating Cloth and Other Deformable Objects. *SIGGRAPH'95*, p.137-144. [doi:10. 1145/218380.218432]
- Wang, C.C.L., Chang, T., Yuen, M.M.F., 2003a. From laser-scanned data to feature human model: a system based on fuzzy logic concept. *Computer-Aided Design*, **35**(3): 241-253. [doi:10.1016/S0010-4485(01)00209-3]
- Wang, C.C.L., Wang, Y., Yuen, M.M.F., 2003b. Feature based 3D garment design through 2D sketches. *Computer-Aided Design*, **35**(7):659-672. [doi:10.1016/S0010-4485(02)00 091-X]
- Wang, J., Lu, G.D., Tang, J.R., 2004. The selection method of the graphics at 3D graphic browser. *Machinery*, **7**:29-33, (in Chinese).
- Yan, M.J., Ning, T., Zhang, Z.P., 2005. Implementation of automating information extraction from 3D body data points in garment CAD. *J. Eng. Graph.*, **26**(1):26-29 (in Chinese).
- Zhang, D.L., Yuen, M.M.F., 2001. Cloth simulation using multilevel meshes. *Computers & Graphics*, **25**(3):383-389. [doi:10.1016/S0097-8493(01)00062-0]