INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN
ICED 03 STOCKHOLM, AUGUST 19-21, 2003

GARSKETCH3D: A SKETCH-BASED 3D APPAREL PRODUCT MODELLING PLATFORM

Charlie C. L. Wang Yu Wang Matthew M. F. Yuen

Abstract

This paper presents a sketch-based platform for intuitively modeling a 3D apparel product around a 3D human model with pre-defined features. After briefly introducing the non-manifold data structure utilized in our system, three techniques are reviewed to obtain a feature human model from a laser scanner, a size table, or photographs. With the feature human model, the user can interactively draw several 2D freeform strokes on the screen to construct a 3D garment by specifying either the boundary of a garment on the surface of a human body (for tight clothes), or the profile of a garment template related to the feature human model (for relatively loose clothes). Our platform supports two modification operations using 2D strokes to modify the constructed garment in the 3D space; they are mesh extrusion and mesh cutting. Finally, after cutting the modified garment into several pattern components, each component is flattened into 2D patterns for the garment manufacturing processes. Our system provides an intuitive and efficient platform to create and modify apparel product patterns directly in the 3D space; and the geometry coverage includes closed surfaces, open surfaces, and more complex non-manifold objects.

Keywords: Virtual prototyping, product modelling, virtual design, computer-aided design

1. Introduction

The work presented in this paper comes from the project of building a 3D sketch-based apparel product development platform, whose purpose is to help apparel enterprises to stay ahead of current technology innovations promoted by major retailers and manufacturers. Currently, 2D CAD systems are widely used in the garment industry. It is expected that three-dimensional tools might be used for garment design to improve the efficiency of pattern generation and for more attractive design presentation. It has been estimated that designers wish to design patterns directly on a 3D human model. At present, there are some systems [1-3] simulating 3D dressing results from 2D patterns input, but they do not focus on 3D design for pattern generation. Thus, a design platform that allows the pattern designers to design and modify cloth pattern prototypes in the 3D space is imperative.

Several projects for directly design garment patterns in the 3D space are in process. Sung and Tae [4] is developing a system to generate patterns from 3D body scan data. Their system is based on the stereovision technique to capture the grid drawing on a dummy. Thus, relatively loose clothes cannot be generated. Also, there is a project of US National Textile Center [5] proposing to automatically generate garment patterns from the 3D body scan data. However, their systems are rule-based; they do not have enough flexibility for designers. Different from the above systems, our platform provides several design tools for designers to develop apparel products around human bodies in the 3D space.

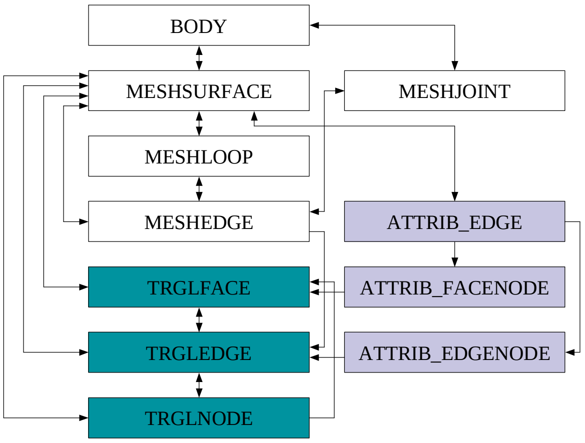
Studies have shown that most designers still prefer to express their creative design idea through 2D strokes. It is important for a computer-aided design system to allow sketched input. Recently, the SKETCH system [6] introduced a gesture-based interface for the rapid modeling of CSG-like models consisting of simple primitives, and the Teddy system [7] presented a sketching interface for quickly and easily designing freeform models such as stuffer animals and other rotund objects. This paper extends their ideas to develop several tools for modeling a 3D garment around a 3D human model through 2D sketches. The geometry coverage of our system is extended from closed surfaces (the SKETCH system and the Teddy system only cover these) to open surfaces, or even more complex non-manifold objects. Our platform can be implemented on a standard PC set (preferably equipped with pen pad) or a tablet PC set at present; and a handheld or palmtop PC set is another piece of equipment that has potential in the future.

After introducing the non-manifold data structures, we briefly review three different techniques to build a human model from a laser scanner, a size chart, or photographs. Two sketch-based design operations are presented in section 4 to construct a tight 3D garment by specifying the boundary of a garment on the surface of a human body (skinning), or to construct a relatively loose 3D garment by specifying the profile of a garment template related to the feature human model (construction with sketched 2D profiles). In section 5, we support the mesh cutting tool and the mesh extrusion tool to modify the constructed 3D garment using 2D strokes. Finally, after cutting the modified garment into several pattern components, each component is flattened into 2D patterns for the garment manufacturing processes.

2. Non-manifold data structure

In the design processes and engineering analyses, design abstractions expressed as lines or surfaces are integral parts of the conceptual model for a physical object. This conceptual modeling approach can be equally applied to non-physical objects. In geometric modelling, such an approach is commonly referred as *non-manifold geometric modeling* or *non-homogeneous geometric modeling* in view of the nature of the modeling domain. Since a 3D garment is usually presented by a set of jointed stand-alone mesh surfaces, a 3D apparel product modeling system should be able to construct both manifold and non-manifold surfaces. Therefore, a data structure [8], which combines the boundary representation and the complex-based representation, is utilized.

The framework of the data structure is shown in Figure 1. An object stored by this data structure is a collection of MESHSURFACEs, each of which is a complex of triangles. The MESHSURFACEs are jointed by a new entity – MESHJOINT, which is comprised of ordered MESHEDGEs on the MESHSURFACEs. Locally, this data structure is a little bit similar to the *winged-edge* data structure. Every TRGLEDGE contains the link to its left face and right face; and every TRGLNODE has its adjacent faces, edges, and nodes stored. Four attributes are also defined in the data structures for the representation of features. They include ATTRIB_NODE, ATTRIB_EDGE, ATTRIB_EDGENODE, and ATTRIB_FACENODE. The ATTRIB_EDGENODE and ATTRIB_FACENODE both are attribute nodes; they are derived from ATTRIB_NODE. An ATTRIB_EDGENODE is an attribute node on the TRGLEDGE, its coordinate depends on the position of TRGLEDGE's nodes; and an ATTRIB_FACENODE is an attribute node in the TRGLFACE, its coordinate depends on the position of TRGLFACE's nodes. An ATTRIB_EDGE is an ordered collection list of ATTRIB_NODEs, which can be either ATTRIB_EDGENODEs or ATTRIB_FACENODEs.



BODY	Entire Model Complex	MESHSURFACE	Mesh Surface
MESHLOOP	Mesh Surface Loop	TRGLFACE	Triangular Face
MESHEDGE	Mesh Surface Edge	TRGLEDGE	Triangular Face Edge
MESHJOINT	Mesh Surfaces Joint	TRGLNODE	Triangular Face Node
ATTRIB_EDGE	Curves on Mesh Surface	ATTRIB_FACENODE	Nodes on Triangular Face
ATTRIB_EDGENODE	Nodes on Triangular Edge		

Figure 1 Data structure framework by Wang [8]

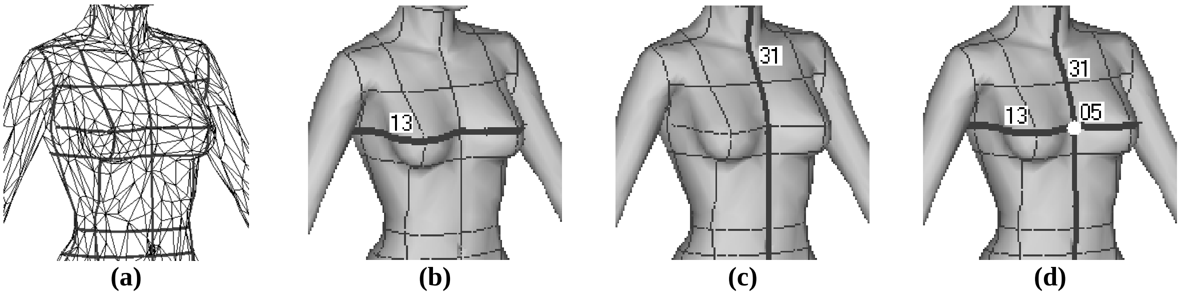


Figure 2 Feature lines and points on a human model

3. Pre-processing: feature human construction

Since 3D apparel products are designed around 3D feature human models, the feature human models should firstly be prepared for the platform. A feature human model can be built from a laser scanner [9], a size chart [10], or photographs [11]. In our semantic feature representation of a human model, each semantic feature curve consists of a sorted set of line segments lying on the mesh surface of a reference model. The line segments might not be the topological edges on the reference model; they can pass through triangular faces (see Figure 2a). The feature curves are stored by the ATTRIB_EDGEs in the data structure (given in section 2). The line segments that belong to a specific semantic feature curve have a common feature ID number. Each semantic feature point is an intersection point of two semantic feature curves, and is stored by the ATTRIB_NODE. Every semantic feature point has its own ID number. For example, in Figure 2b, the bold line segments with ID number 13 represents the chest feature curve of a human model; in Figure 2c, the bold line segments with ID number 31 represents the center-front feature curve of a human model; and in Figure 2d, the white point is the feature point determined by these two feature curves, the ID number of the feature point is 05.

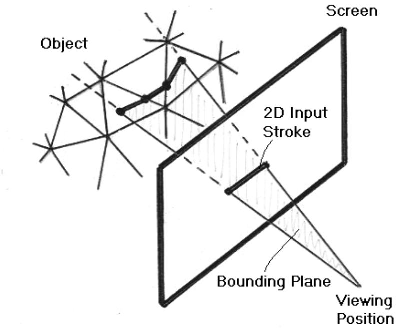


Figure 3 Specify boundaries

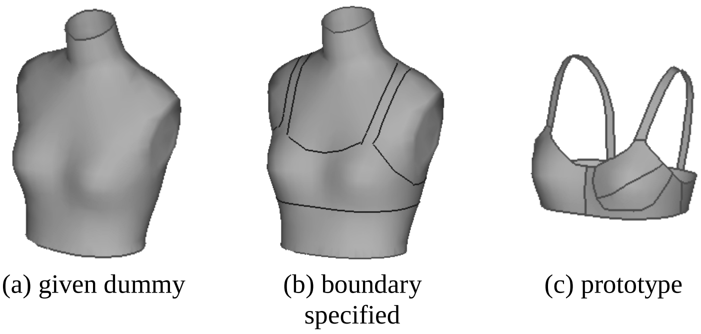


Figure 4 Feature lines and points on a human model

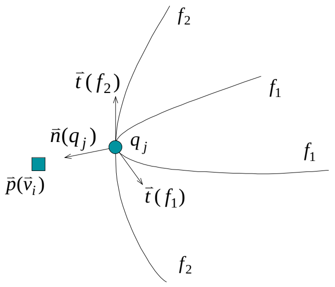


Figure 5 Feature node encoding

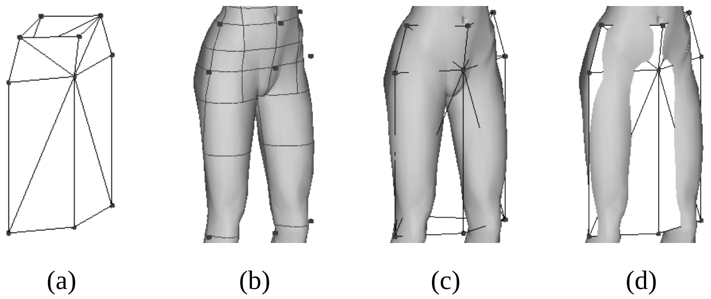


Figure 6 Construct the topological graph interactively

4. Design operations

4.1 Tight clothes

For tight fitting clothes, the user can interactively draw 2D freeform strokes on the surface of a given model to specify the 3D boundaries of a garment (Figure 4). Each line segment of the input 2D strokes is projected on to the mesh surface of the human model along the view direction to create the boundaries. The overall procedure is: for each line segment of the 2D stroke, first determine a bounding plane containing the projection of the line segment from the viewing position; then the system finds all intersections between the plane and each polygon of the object, and splices the resulting 3D line segments together (see Figure 3). The spliced 3D line segments are stored by the ATTRIB_EDGEs in the data structure. The projected sketched boundaries pass over the triangles of the mesh surface of a given human model. Firstly the constrained Delaunay triangulation algorithm [12] is applied to convert boundary curves into triangular edges of the given 3D mesh; and then the triangles surrounded by the boundary curves are copied out to form a new open surface (Figure 4b and 4c). This open surface is a prototype 3D garment model. We usually use this operation to construct tight fitting underwear or swim suits. This sort of clothes is always created on the 3D dummies whose 3D surfaces have already included the offset between the human body and the clothes

4.2 Relative loose clothes

For relative loose clothes, the user interactively draw 2D freeform strokes to specify the profiles of a garment template related to the feature human model [13]. After that, the surface of a prototype garment interpolating the profiles is constructed by a subdivision approach. The constructed garment patterns are related to the features on a human model, so the patterns can be regenerated automatically when creating the same style of garment for other human models.

The structure of a garment constructed by this method is represented in a complex of triangular mesh surfaces M^f whose nodes are related to the features on human models. The relationship between M^f and a feature based human model is built by the following feature-node encoding step. Suppose there is a feature node $\vec{v}_i$ in the garment template and a feature point q_j on a human model H, where j is the ID number of point q_j (described in section 3). The position $\vec{p}(q_j)$ of q_j is determined by the intersection of two feature curves f_1 and f_2. The vectors $\vec{t}(f_1)$, $\vec{t}(f_2)$, and $\vec{n}(q_j)$ form a local coordinate frame as shown in Figure 5. The three vectors may not be perpendicular to each other, but at least $\left(\vec{t}(f_1)\,\vec{t}(f_2)\,\vec{n}(q_j)\right)\neq 0$. Thus, the position $\vec{p}(\vec{v}_i)$ of any vertex $\vec{v}_i \in V^f$ can be represented by

$$\vec{p}(\vec{v}_i) = \vec{p}(q_j) + \alpha_i \vec{n}(q_j) + \beta_i \vec{t}(f_1) + \gamma_i \vec{t}(f_2) \tag{1}$$

The encoding process that relates one feature node $\vec{v}_i \in V^f$ to a feature point q_j on H is actually a process to determine the four elements - $(j, \alpha_i, \beta_i, \gamma_i)$ of vertex $\vec{v}_i$. This node encoding method strongly preserves the orientation of garment feature nodes to a human model. After the positions of feature nodes are defined, we should determine the topological graph K^f of M^f. The topological graph K^f consists of the adjacency of the vertices, edges and faces. The K^f of a garment feature template is constructed by interactive tools. For example, to construct a feature template for a skirt likes Figure 6a, a set of feature nodes are created and encoded (Figure 6b); then the edges and faces linking the feature nodes are interactively input (Figure 6c and 6d).

After the coarse shape of the feature template is determined, 2D sketches are used to specify the 3D profiles of some triangular edges in the feature template, which describe the detail shape of the constructed object and are interpolated in the shape construction step (Figure 7). The 3D profile of a triangular edge is a 3D curve whose two endpoints coincide with the two endpoints of the edge. For a given feature edge $\vec{v}_i\vec{v}_j$ with endpoints $\vec{v}_i$ and $\vec{v}_j$, it is selected by the stroke Ψ if $\vec{v}_i\vec{v}_j$ is visible and the distances between $\zeta(\vec{v}_i)$, $\zeta(\vec{v}_j)$ and Ψ is less than ε, where $\zeta : \Re^3 \rightarrow \Re^2$ is the map that sends spatial point $\vec{v}_i \in \Re^3$ to screen point $\psi_i \in \Re^2$, and ε is a small tolerance value (e.g., $\varepsilon = 4$ pixels). In the following, a plane is determined to project the points $\psi_i \in \Psi$ to convert them into 3D points $\psi *_i \in \Psi *$ in $\Re^3$, where $\Psi *$ is a list of 3D points. When multiple feature edges are selected, a plane that approximately passes through the selected feature edges is conducted. If only one feature edge is selected, we use the plane that bisects the dihedral angle along the chosen edges to project the points $\psi_i \in \Psi$. In this way, the sketched profile faces towards the camera as much as possible. After all $\psi_i \in \Psi$ are converted to $\psi *_i \in \Psi *$ in $\Re^3$, we separate them into intervals and store the points in one interval in its related edge. For a selected feature edge $\vec{v}_l\vec{v}_m$, we search the closest point $\psi *_l$ to $\vec{v}_l$ in $\Psi *$, and the closest point $\psi *_m$ to $\vec{v}_m$ in $\Psi *$. Then, the points $\psi *_{j,\, j=l,\ldots,m}$ are stored as the attached points list in $\vec{v}_l\vec{v}_m$. Two examples of specifying profiles through a 2D stroke is given in Figure 8.

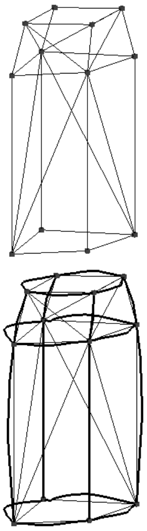
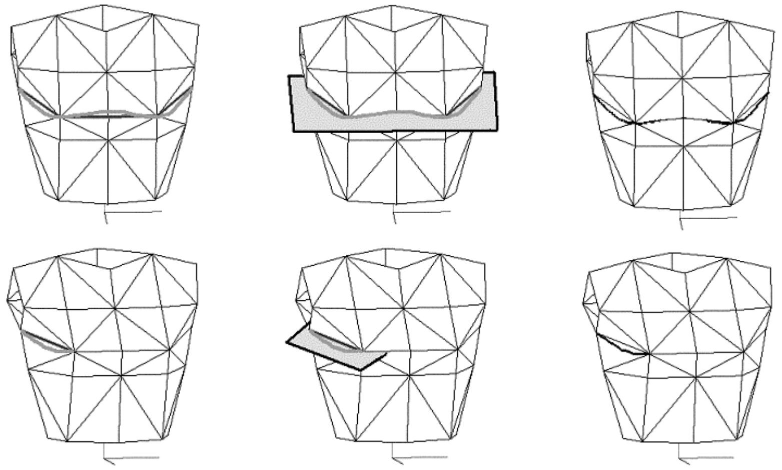
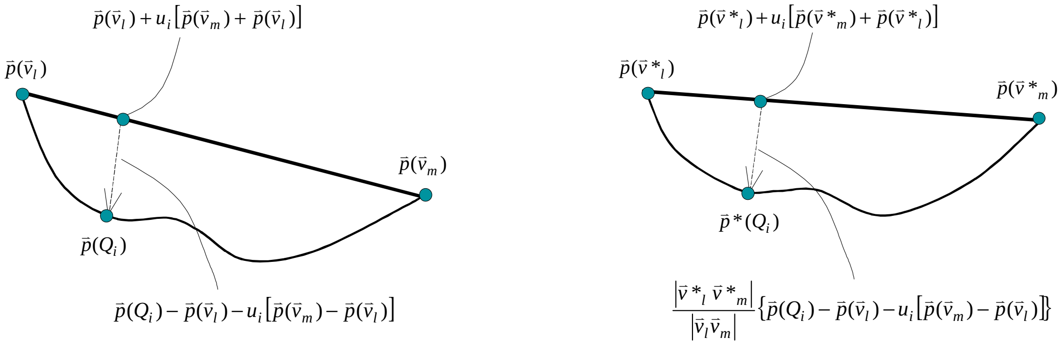


Figure 7 With profiles



(a) (b) (c)

Figure 8 Specify profile through a 2D stroke



$\vec{p}(\vec{v}_l)+u_i[\vec{p}(\vec{v}_m)+\vec{p}(\vec{v}_l)]$
$\vec{p}(\vec{v}_l)$
$\vec{p}(\vec{v}_m)$
$\vec{p}(Q_i)$
$\vec{p}(Q_i)-\vec{p}(\vec{v}_l)-u_i[\vec{p}(\vec{v}_m)-\vec{p}(\vec{v}_l)]$
$\vec{p}(\vec{v}*_l)+u_i[\vec{p}(\vec{v}*_m)+\vec{p}(\vec{v}*_l)]$
$\vec{p}(\vec{v}*_l)$
$\vec{p}(\vec{v}*_m)$
$\vec{p}*(Q_i)$
$\frac{|\vec{v}*_l\,\vec{v}*_m|}{|\vec{v}_l\vec{v}_m|}\{\vec{p}(Q_i)-\vec{p}(\vec{v}_l)-u_i[\vec{p}(\vec{v}_m)-\vec{p}(\vec{v}_l)]\}$

(a) (b)

Figure 9 Specify profile through a 2D stroke

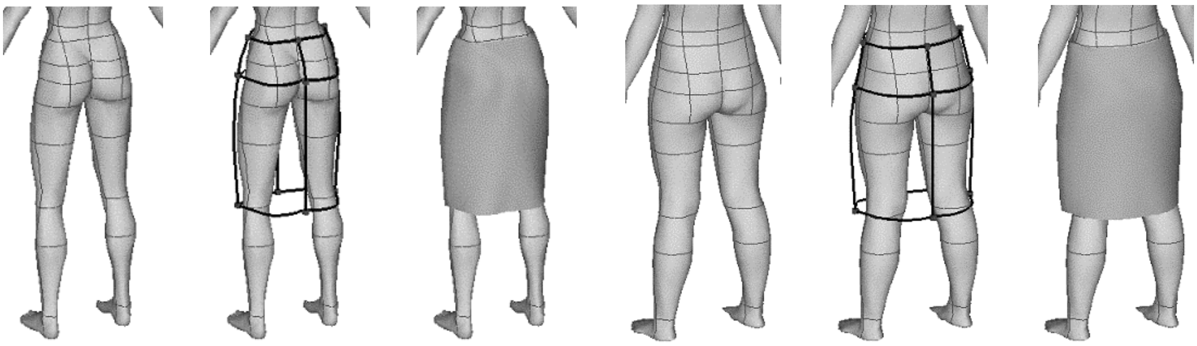
When re-generating the 3D object on a reference object with the different shape, the positions of the feature nodes in the feature template are changed. We need to shift the position of the attached points on triangular edges to re-generate the "new" 3D profiles. If an edge is moved from $\vec{v}_l\vec{v}_m$ to $\vec{v}*_l\,\vec{v}*_m$, the new positions of points $Q_{i,\,i=0\ldots n-1}$ attached to it are shifted by scaling the vector between them and $\vec{v}_l\vec{v}_m$. The idea is shown in Figure 9, where the vector between Q_i and $\vec{v}_l\vec{v}_m$ is scaled by $|\vec{v}*_l\,\vec{v}*_m|/|\vec{v}_l\vec{v}_m|$. The formulation to compute the new positions of Q_i is shown below

$$\vec{p}*(Q_i)=\frac{|\vec{v}*_l\,\vec{v}*_m|}{|\vec{v}_l\vec{v}_m|}\{\vec{p}(Q_i)-\vec{p}(\vec{v}_l)-u_i[\vec{p}(\vec{v}_m)-\vec{p}(\vec{v}_l)]\}+\vec{p}(\vec{v}*_l)+u_i[(\vec{p}(\vec{v}*_m)-\vec{p}(\vec{v}*_l)] \qquad (2)$$

where u_i is the parameter defined by the lengths of line segments [14].

A triangular mesh surface interpolating the specified 3D profiles is constructed by applying a variational subdivision scheme on the given garment template. The basic idea of a variational subdivision scheme is to iteratively applying a topological splitting operator to introduce new vertices to increase the number of degree of freedom, followed by a discrete fairing operator to increase the overall smoothness [15]. The surface constructed in our approach is around a 3D human model, so the collision between the constructed surface and the human model may happen during the surface construction procedure. Here, we modified the variational subdivision scheme by integrating the collision avoidance. In order to achieve a fast scheme, we implement the image-based collision detection [16].

The constructed garment surfaces are related to the features on a human model, so the surface can be regenerated automatically when using another human model. For example, after we specify profiles on the human model H_1 in Figure 10a, we obtain the garment template mesh M_0^f in Figure 10b. The constructed surface M_∞^f is shown in Figure 10c. When we construct the same style of skirt for the human model H_2 in Figure 10d, we relocate the feature nodes in M_0^f according to the features on H_2 by eq. (1) and shift the attached points by eq. (2), the new garment template mesh M_0^* is automatically determined (Figure 10e). After mesh refinement, its related surface is obtained as shown in Figure 10f



(a) H_1 (b) M_0^f (c) M_∞^f (d) H_2 (e) M_0^f (f) M_∞^f

Figure 10 Apply the same garment template on different human models

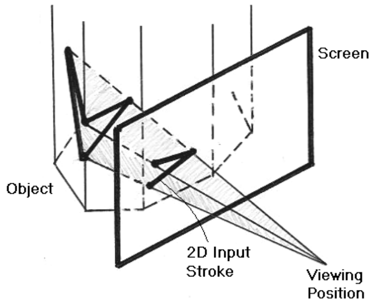


Figure 11 Mesh cutting illustration

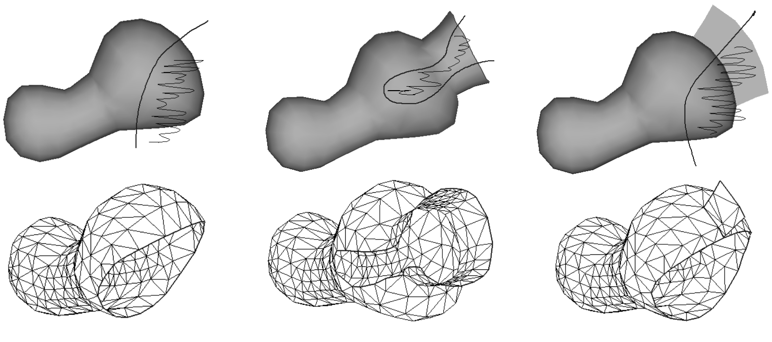
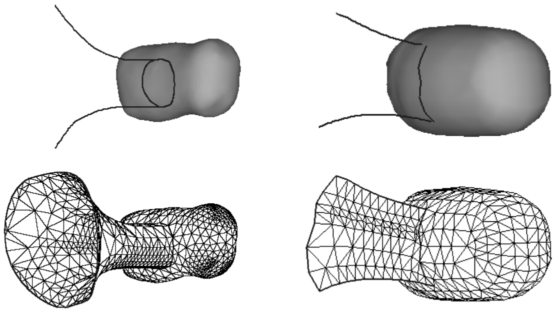
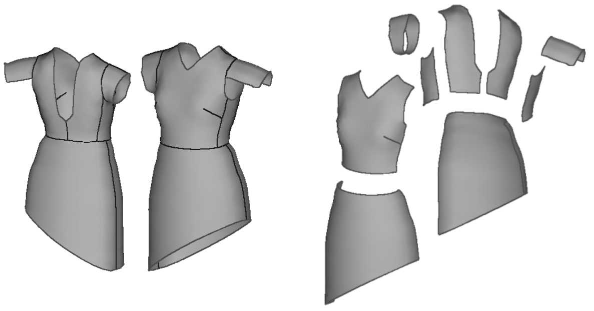


Figure 12 Examples of mesh cutting



(a) (b)

Figure 13 Mesh extrusion



(a) (b)

Figure 14 Separating

5. Modification operations

5.1 Cutting

The mesh cutting operation is to remove some parts of the given mesh surface by input 2D strokes. Similar to the Teddy system [7], the cutting operation is based on the painting algorithm. After painting a curve on the surface of a model, the constrained Delaunay triangulation algorithm [12] makes the painted curves to form the triangle edges of the model. Removing the triangles on the user-selected side of the painted curve (specified by another stroke) from the model, the cutting result is obtained (illustrated in Figure 11 and 12).

5.2 Extrusion

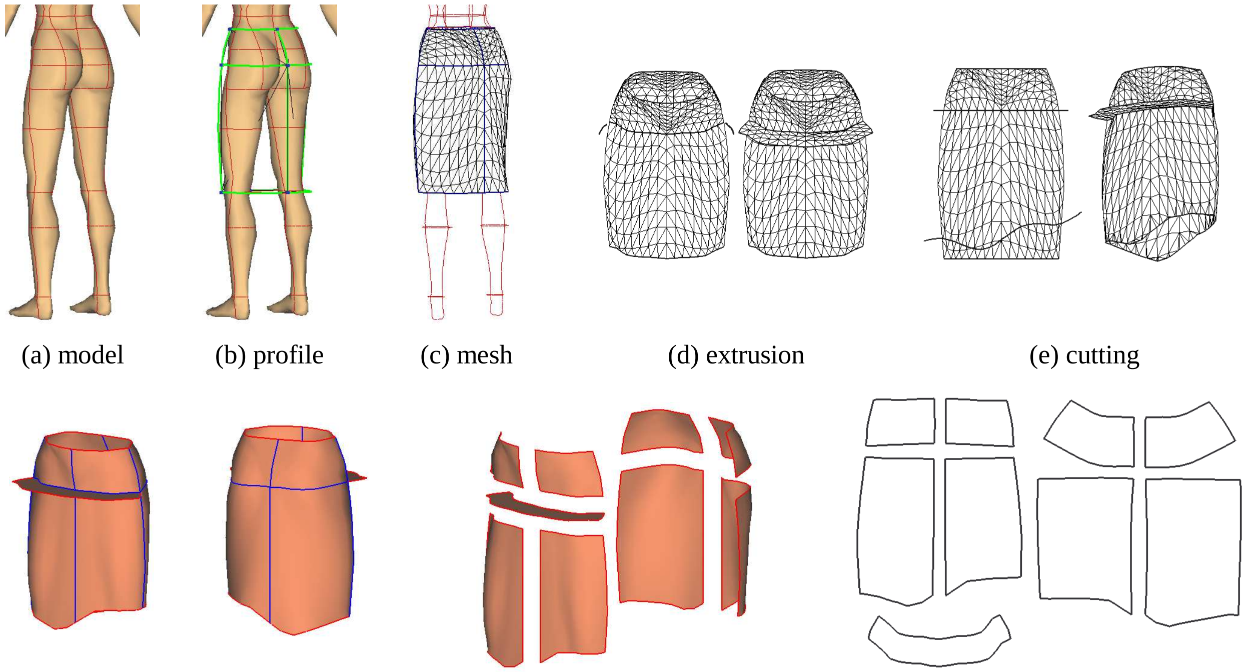
The extrusion operation is applied in our system to create new polygonal meshes based on base surface line segments (called the base curve) and extruding strokes. The mesh extrusion method implemented here is based our previously development [17]. The mesh extrusion method is best illustrated by examples such as that shown in Figure 13. In these examples, the given initial model is a closed mesh surface; the approach allows the user to sketch 2D input strokes (a stroke on the surface of given mesh, and 2D strokes depicting the profile curves of the extruded surface) to extrude a surface from the given mesh. Firstly the user draws a stroke on the object surface; then the user rotates the model to bring the stroke sideways and draw silhouette lines to extrude the surface. A sweep operation is applied to construct the 3D shape by moving the surface base curve, which is obtained by projecting the first stroke onto the surface of the given mesh, along the skeleton of the profile curves. The implementation algorithm is described as follows.

5.3 Separating

The same as the cutting tool, the separating operation is also based on the painting algorithm. After painting the separating curves on the surface of a model (Figure 14a), we apply the constrained Delaunay triangulation algorithm [12] to convert the painted curves into triangle edges of the model. After re-triangulation, the whole model is divided to several sets of triangles; each set of triangles is a triangular mesh, which is a component of the product model (see Figure 14b).

6. Post-processing: mesh surface flattening

Using the operations introduced in section 4, the user can create and modify 3D apparel models intuitively and efficiently. However, the garment manufacturing industry needs 2D patterns to be used in the manufacturing processes. Thus, we implement the energy-based surface-flattening algorithm [18] to generate the corresponding 2D patterns of a 3D apparel model by using a spring-mass model. This procedure consists of triangles flattening and planar mesh deformation. During the triangles flattening phase, triangles are flattened one by one; and a partial spring-mass system containing flattened triangles is deformed to release the strain energy during the flattening. After all the triangles are flattened, the spring-mass system will have all the triangles of the given surface. The planar triangular mesh deformation process is directed by the energy function of the spring-mass system. By releasing the energy function, we can obtain the 2D pattern related to the given 3D mesh surface.


(a) model (b) profile (c) mesh (d) extrusion (e) cutting

(f) parting curves defined (g) 3D pieces (h) related 2D patterns

Fig. 25 Design a skirt in GarSketch3D

7. Experimental Result

The experimental result of designing a skirt using our GarSketch3D system is shown in Figure 15. From Figure 15a to 15c, a skirt is constructed from a feature template on a human model with pre-defined features. After extruding and cutting in Figure 15d and 15e, the final skirt can be separated (Figure 15f and 15g) and flattened into 2D patterns (Figure 15h).

8. Conclusion

This paper describes the development of a sketch-based platform for designing 3D apparel products around a three-dimensional human model with features defined. Our prototype platform is implemented on Windows NT, using Visual C++ and OpenGL Library. The mesh construction and modification process can be completed in real-time on a standard PC, but a short pause is experienced (a few seconds) when the model becomes complex. Compared with other 3D apparel product designs and simulation platforms, our system provides an intuitive and efficient platform to create and modify apparel product patterns directly in the three-dimensional space. The spherical topology limitation of other sketch-based 3D freeform modeling approaches is overcome.

Our algorithms currently may fail or generate unintuitive result when the user draws unexpected strokes, so in future research, we will enhance our algorithms to be more robust and flexible to handle a variety of user input. The models created in our system are relatively simple now. Further research can also focus on how to enhance the ability of modeling complex model by sketched input.

Reference

[1] Volino P., Courchesne M., and Thalmann N.M., “Versatile and efficient technique for simulating cloth and other deformation objects”, SIGGRAPH 95 Proceeding, ACM., pp.137-144, 1995, New York, USA.

[2] “DressingSim Pattern Maker”, http://www.dressingsim.com, 2001.

[3] “Maya Cloth”, http://www.aliaswavefront.com/en/products/maya/unlimited.shtml, 2002.

[4] Sung M.K., and Tae J.K., “Garment pattern generation from body scan data”, Computer-Aided Design, vol.35, no.7, pp.611-618, 2003.

[5] National Textile Center Project I00-S15, “Automated garment development from body scan data”, http://www.tx.ncsu.edu/3dbodyscan/, 2002.

[6] Zeleznik R.C., Herndon K.P., and Hughes J.F., “SKETCH: An interface for sketching 3D scenes”, SIGGRAPH 96 Proceeding, 1996, ACM., pp.163-170, New York, USA.

[7] Igarashi T., Matsuoka S., and Tanaka H., “Teddy: a sketching interface for 3D freeform design”, SIGGRAPH 99 Proceeding, 1999, ACM., pp.409-416, New York, USA.

[8] Wang C.C.L., “Sketch based 3D freeform object modeling with non-manifold data structure”, Ph.D. Thesis, 2002, Hong Kong University of Science and Technology.

[9] Wang C.C.L., Chang T.K.K., and Yuen M.M.F., “From laser-scanned data to feature human model: a system based on fuzzy logic concept”, Computer-Aided Design, vol.35, no.3, pp.241-253, 2003.

[10] Chang T.K.K., Wang C.C.L., and Yuen M.M.F., “Web-based design and manufacturing of custom manniquin model”, ICED01 - The 13th International Conference on Engineering Design, 2001, SECC, Glasgow, UK.

[11] Wang C.C.L., Wang Y., Chang T.K.K., and Yuen M.M.F., “Virtual human modeling from photographs for garment industry”, Computer-Aided Design, vol.35, no.6, pp.577-589, 2003.

[12] de Floriani, L, and Puppo, E., “An online algorithm for constrained delaunay triangulation”, CVGIP-Graphical Models & Image Processing, vol.54, no.4, pp.290-300, USA.

[13] Wang C.C.L., Wang Y., and Yuen M.M.F., “Feature based 3D garment design through 2D sketches”, Computer-Aided Design, vol.35, no.7, pp.659-672, 2003.

[14] Piegl L., and Tiller W., “The NURBS Book (2nd ed)”, 1997, Berlin; Hong Kong: Springer.

[15] Kobbelt L., “Discrete fairing and variational subdivision for freeform surface design”, The Visual Computer, vol. 16, no. 3/4, pp. 142-158, 2001.

[16] Vassilev T., Spanlang B., and Chrysanthou Y., “Fast cloth animation on walking avatars”, Eurographics 2001 Proceeding, 2001.

[17] Wang C.C.L., and Yuen M.M.F., “Freeform extrusion by sketched input”, Computers & Graphics, vol. 27, no.2, 2003.

[18] Wang C.C.L., Smith S.S.F., and Yuen M.M.F., “Surface flattening based on energy model”, Computer-Aided Design, vol.34, no.11, pp.823-833, 2002.