Parameterization and Parametric Design of Mannequins

Charlie C. L. Wang

Department of Automation and Computer-Aided Engineering, Chinese University of Hong Kong,

Shatin, N.T., Hong Kong

E-mail: cwang@acae.cuhk.edu.hk; Tel: (852) 2609-8052; Fax: (852) 2603-6002

Abstract

This paper presents a novel feature based parameterization approach of human bodies from the unorganized cloud points and the parametric design method for generating new models based on the parameterization. The parameterization consists of two phases. Firstly, the semantic feature extraction technique is applied to construct the feature wireframe of a human body from laser scanned 3D unorganized points. Secondly, the symmetric detail mesh surface of the human body is modeled. Gregory patches are utilized to generate G^1 continuous mesh surface interpolating the curves on feature wireframe. After that, a voxel-based algorithm adds details on the smooth G^1 continuous surface by the cloud points. Finally, the mesh surface is adjusted to become symmetric. Compared to other template fitting based approaches, the parameterization approach introduced in this paper is more efficient. The parametric design approach synthesizes parameterized sample models to a new human body according to user input sizing dimensions. It is based on a numerical optimization process. The strategy of choosing samples for synthesis is also introduced. Human bodies according to a wide range of dimensions can be generated by our approach. Different from the mathematical interpolation function based human body synthesis methods, the models generated in our method have the approximation errors minimized. All mannequins constructed by our approach have consistent feature patches, which benefits the design automation of customized clothes around human bodies a lot.

Keywords: human body, sizing dimensions, 3D scan data, feature-based modeling, and fashion industry.

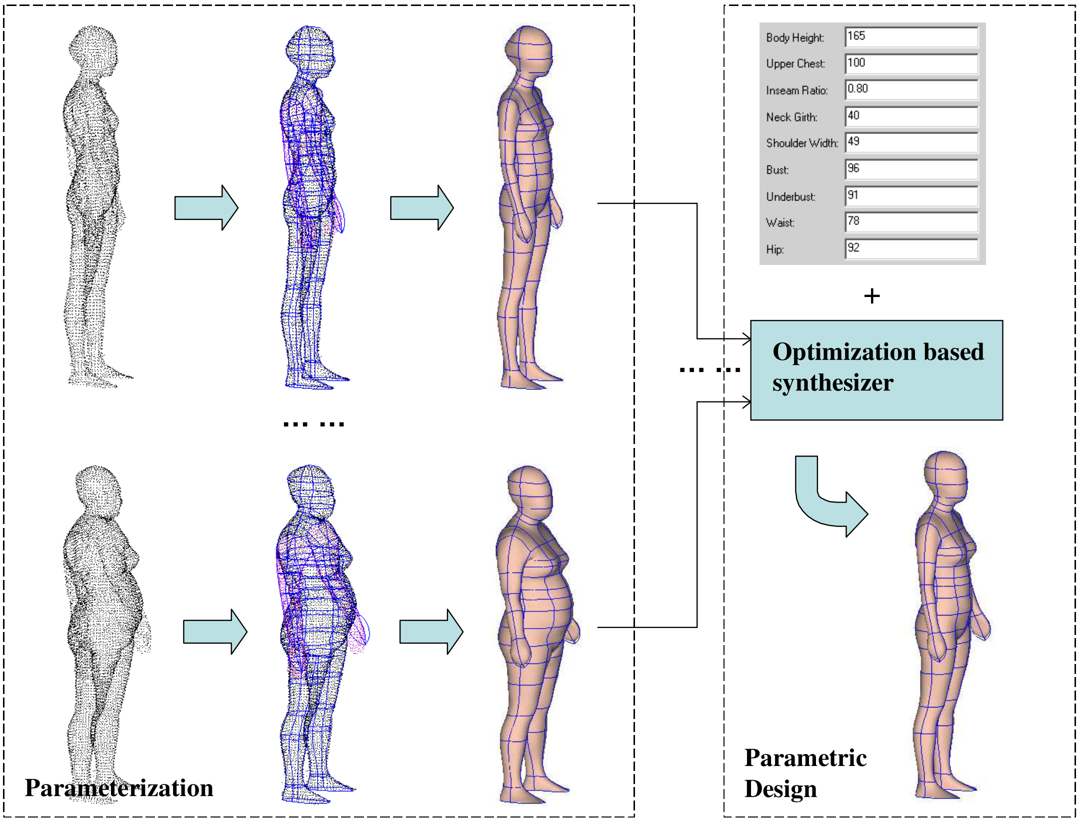
1. Introduction

In this paper, an approach for the parametric design of human models is developed. The modeled human bodies are parameterized for the following design automation of clothes around them. As a first step, we collect a database containing the measurements of a representative number of people. These measurements are gathered from scanners that generate point cloud data. The point cloud data is analyzed to build a wireframe mesh, which is then fleshed out into a complete surface model for the human torso using Gregory patches to interpolate curves in the wireframe. The complete mesh surface is further updated according to the scanned points by a voxel-based approach. Since the complete mesh surface is with a fixed structure and connection to represent a mannequin, every human model reconstructed in this way has been parameterized. After the database is finished, the ability of generating 3D parameterized human models from input dimensions is provided – no scanning is needed. The parameterization of human models is a necessary step for developing a design automation system of apparel products. The parameterized human body by this method gives a point-to-point corresponding among a set of human body surfaces with the same overall structure. This kind of point-to-point mapping provides us the possibility to encode the relationship between clothes and a human body, so clothes around other human bodies can automatically be regenerated by maintaining the same relation.

The work presented in this paper is an extension of constructing feature-based human models for building a 3D digital mannequin database [1]. Feature-based human models are fundamentals for developing the design automation technology of customized clothes. The features extracted from the human model are the major contribution of the database to the fashion industry. In [1-3], the feature curves and points are utilized as the semantic features; however, recent developments find that only feature curves and points are not enough for modeling 3D digital clothes around human models – feature patches are actually required. In this paper, we begin the parameterization with the unorganized cloud points of human bodies obtained from 3D laser scanners. The entire human body is subdivided into a certain number of feature patches interpolating the given cloud points. The feature patches at the same location on different human bodies are correlated, where the detail geometric shape of a human model is given. The sizing parameters are given by feature curves. By the feature entities (feature patches, feature curves, and feature nodes), the geometry of a human body is fully registered and parameterized.

Based on the parameterization of a human model, we can easily obtain its sizing dimensions through feature curves. The parameterized models are stored in a database using the sizing dimensions as searching criteria. After establishing the database, we can construct new models according to user input sizing parameters by synthesizing selected models from the database. This leads to the problems of how to select example models and how to synthesize the models. In our approach, we select a number of the closest example models to the given sizes; then, a numerical optimization approach is applied to compute the weight coefficients of synthesizing the example models. Finally, the requested model is generated by interpolating the examples models with optimized weights. This is named as the parametric design of human bodies

The target example data size of our digital human database is more than 5000 persons. Since human models born in different regions have different morphologic features, the samples from different regions are stored separately. The working flow of our approach is clearly illustrated by Fig.1. The parameterization has two phases: 1) the registration of feature wireframe – this is based on the feature extraction technique (the object under our consideration are of the same class, so the semantic feature extraction technique [1] is applied); 2) the modeling of surfaces according to feature wireframe and cloud points. After inputting sizing parameters to create a new human model, our synthesizer computes the weight coefficients for interpolation and interpolates selected examples by the weights – the parametric design result is then obtained.



Body Height: 165
Upper Chest: 100
Inseam Ratio: 0.80
Neck Girth: 40
Shoulder Width: 49
Bust: 96
Underbust: 91
Waist: 78
Hip: 92

Fig. 1 The procedure of parameterization and parametric design of human models

The major contributions of this paper are 1) an efficient feature-based parameterization technique for establishing a point-to-point corresponding among a set of human body surfaces with the same overall structure, and 2) a numerical optimization based synthesis technique for constructing a new human body with the by specified sizing dimensions – the resultant model with approximation errors minimized. As the example models are parameterized, the synthetic result is certainly parameterized, which gives great benefits to the design automation of cloth products around human models.

The rest of this paper is organized as follows: after reviewing related works in section 2, section 3 describes the necessary steps for registering a feature wireframe on the unorganized cloud points for a human body. In section 4, Gregory patch is adopted to construct G^1 continuous surface interpolating the feature wireframe, a voxel-based algorithm is applied to add details on the surface, and the human surface is made symmetric. The numerical optimization based synthesis algorithm for the parametric design of human bodies is given in section 5, where the strategy of choosing appropriate example models from database is also described. Finally, in section 6, the application for the design automation of customized clothes, which gains great benefit from the technique presented here, is demonstrated.

2. Literature Review

The human body modeling methodologies in literature can be classified into the creative approaches and the reconstructive approaches. Anatomically based modelers [4, 5] can simulate underlying muscles, bones, and generalized tissue. They fall into the creative category of human modeling approaches. The interactive design is allowed in the anatomy-based modelers; however, these modelers require a relatively slow production time. Recently, a lot of the reconstruction approaches has been investigated to build 3D geometry of human automatically by capturing existing shape [1-2, 6-9]. As mentioned by Seo and Magnenat-Thalmann [10], the disadvantage of these techniques is that it is very difficult to automatically modify the reconstructed models to different shapes following the user intends. Example-based shape modeling technique [10-13] is a good alternative to overcome this disadvantage. Our parametric design algorithm borrows some idea from the example based shape modeling. In the example-based shape modeling, all examples must have the same parameterization. Thus, our approach begins from the parameterization of a human model.

Related to the parameterization of unorganized cloud points, Ma and He [14] presented an approach to shape a single B-spline surface with a cloud of points, their work is further enhanced on fitting a hybrid mathematical model of B-spline surfaces and Catmull-Clark subdivision surfaces to represent objects with general quadrilateral topology [15]; Barhak and Fischer [16] also presented a PDE based method about the parameterization for reconstruction of 3D freeform objects from laser-scanned data. Sienz et al. [17] developed a fitting technique to generate computational geometric models of 3D objects defined in the form of a point cloud. All the above approaches are pure geometry oriented and feature technology does not benefit the mesh construction process. Since human models are feature-based which fall in the same class of objects with features, the recognized features on the scanned cloud points will benefit the surface parameterization and construction process.

Blanz and Veter [11] modeled facial variation using a deformable polygonal mesh. Their idea is to create a single surface representation that can be adapted to fit all of the example faces. Each vertex's position and color may vary between examples, but its semantic identity must be the same. Therefore, the main challenge in constructing the deformable face is to re-parameterize the example surfaces so that they have a consistent representation – they adopted the 2D optical flow to achieve that. However, in the case of whole human body models, it becomes more difficult since the whole body cannot be parameterized cylindrically. To solve this problem, Allen et al. [9] developed a fitting method to register high-resolution template meshes to detailed human body range scans with sparse 3D markers. Their approach is based on numerical optimization computing, which is usually time-consuming as the number of variables to be determined is three times the number of vertices on their high-resolution template mesh. Au and Yuen [18] discussed the issues of applying feature technology to the reverse engineering of a mannequin. In their approach, the feature model of a mannequin consists of the major features of the torso for garment design. Fitting the generic feature model to the point cloud yields the mannequin feature model of a specific person. This is achieved by optimizing the distance between the point cloud and the feature surface, subject to the continuity requirements. The process of their surface fitting is also very time consuming. Different from [9, 13], the parameterization approach introduce here is based on the semantic feature extraction. After detecting key feature points on the oriented cloud points of the human models, the feature wireframe and the feature patches are constructed successively. The computing time is greatly shortened in our approach as the features are utilized during parameterization.

In the feature based modeling field, object semantics are systematically represented for a specific application domain; in other words, a semantic feature is an application-oriented feature defined on geometric elements. In our approach, the semantic features are specific parts on a human body (e.g., waist, chest, hip, thigh, knee, ankle, etc.). There are two approaches for building a feature model [19]: 1) The design by feature approach creates the feature model of an object by composing the available features in a feature library; 2) The feature recognition approach recognizes various feature from a geometric model of an object according to the feature templates defined in a feature library. This paper follows the second approach to extract features on the scanned cloud points of a human model. Related researches about freeform feature extraction can be founded in literature (ref. [20-23]). Our feature extraction algorithm borrows some idea from them.

In the approach of Seo and Magnenat-Thalmann [10], they adopted the radial basis functions (RBF) to interpolate scattered human examples. The final interpolation of linear combined radial base functions matches the examples exactly. However, if the specified sizing parameters has no exactly matched human model among all the examples, the interpolation function only gives an approximated human model of the input sizing parameters. The approximation errors are not controlled. DeCarlo et al. [15] showed the application of constrained optimization on building an anthropometric face model. Our parametric design algorithm is also a numerical optimization based approach, but the optimized variables are not the position of each vertex – the weight coefficients of synthesis are optimized. Different from example based modeling methods [10-13], this optimization based synthesis approach minimizes the approximation error. Accurate results are given if appropriate examples are chosen. If inappropriate examples are utilized (e.g., the specified sizing dimensions are out of the size range among all examples), our approach can still determine a human model with the minimized approximation error comparing to the given dimensions.

3. Feature Wireframe Construction

The input cloud points in our approach is assumed to have no noisy points, and have its orientation fixed – facing the x direction. During feature extraction, the x-axis is defined to point out of the screen, the y-axis is horizontally pointing to the right in the screen plane, and the z-axis is vertically pointing upwards in the screen plane. The raw scans have to be filtered and reoriented to satisfy our input requirements (e.g., using the algorithms presented in [1]). Building the feature wireframe consists of three steps: 1) extracting the key feature points on the cloud points; 2) using anthropometrical rules to determine the semantic feature points; 3) linking all the feature points by feature curves interpolating the cloud points. The linked feature points and the linking feature curves constitute the feature wireframe, the topology of which keeps consistent to all processed human models.

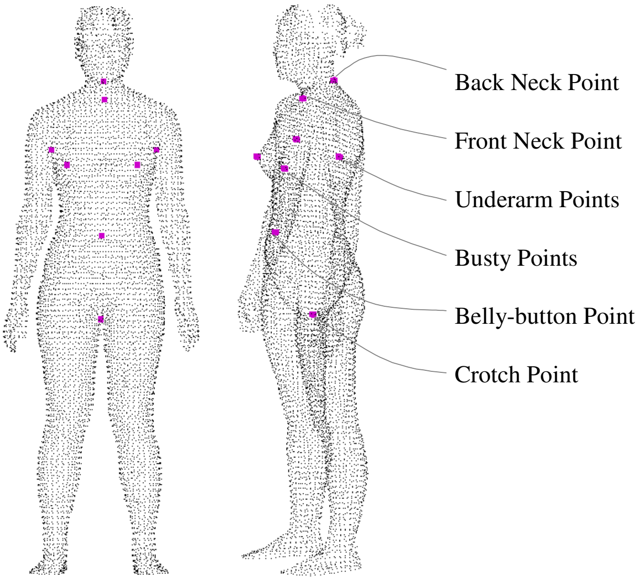


Fig. 2 Key feature points on a human body

3.1 Key feature points

The key feature points on the surface of a human body, including the underarm points, the crotch point, the belly-button point, the front neck point, the back neck point, and the busty points (illustrated in Fig.2), must be extracted first. The fuzzy logic based approach [1] is adopted here. The basic idea is that: using some planes to intersect the 3D unorganized points of a human body's scan or projecting points onto some planes to obtain 2D contours. We can determine turning points on the 2D contours by the "sharp" angles along the contour, where the definition of "sharpness" follows the fuzzy logic concept. On a 2D polygon, if the positions of three adjacent points p_{i-1}, p_i, and p_{i+1} make the value of angle $\angle p_{i-1} p_i p_{i+1}$ smaller than a threshold, we say that p_i is a turning point. With the help of turning points, we can detect the key feature points. The procedures are briefly described as follows.

For the crotch point, cutting the human body from its ½ height downwards, once the intersection break into two circles, the crotch point P_{crotch} is located at the center of the bounding box of two legs' contour (ref. [1]). After projecting the right view of a human body to obtain its silhouette (see the left part of Fig. 3), based on two turning points P_1 and P_2, we have determined the position of the front neck point (by P_1) and the height of the busty points, H_{busty}, (according to P_2). Then, the closest point on the right boundary of the silhouette – P_3 indicates the location of the back neck point. The height from the back neck point to the highest point of scan is the head height, H_{head}. The height of belly-button is usually about the height of crotch point H_{crotch} plus H_{head}. Thus, by cutting several horizontal planes around the height of $H_{crotch} + H_{head}$ (right lower part of Fig. 3), if the turning point P_4 is determined, it is the belly-button point. To determine the exactly busty points, we cut the cloud points at the height of H_{busty} to obtain a contour as shown in Fig. 3, the left lowest point and the right lowest points on the contour are the two busty points. After the height of the belly-button point H_{belly} and the height of the back neck point H_{backneck} are determined, the height of the underarm point H_{underarm} can be computed by an anthropometrical equation: $H_{\text{underarm}}= H_{\text{belly}} +0.55(H_{\text{backneck}} -H_{\text{belly}})$. Therefore, we cut the cloud points there and find the 4 turning points on the contours. The underarm point is in the middle of the front and the back turning points on the cutting plane (see Fig. 3).

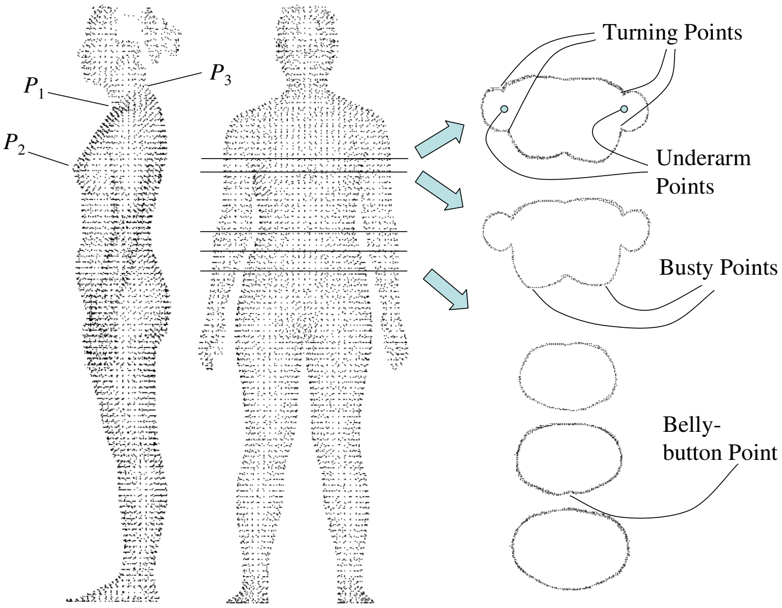


Fig. 3 Determine the key feature points

Based on the above methods, the key feature points can be automatically determined on more than 85% samples. For the 15% examples that lead to the automatic extraction fail, we need to manually locate the key feature points on the scans (which is similar to [9]).

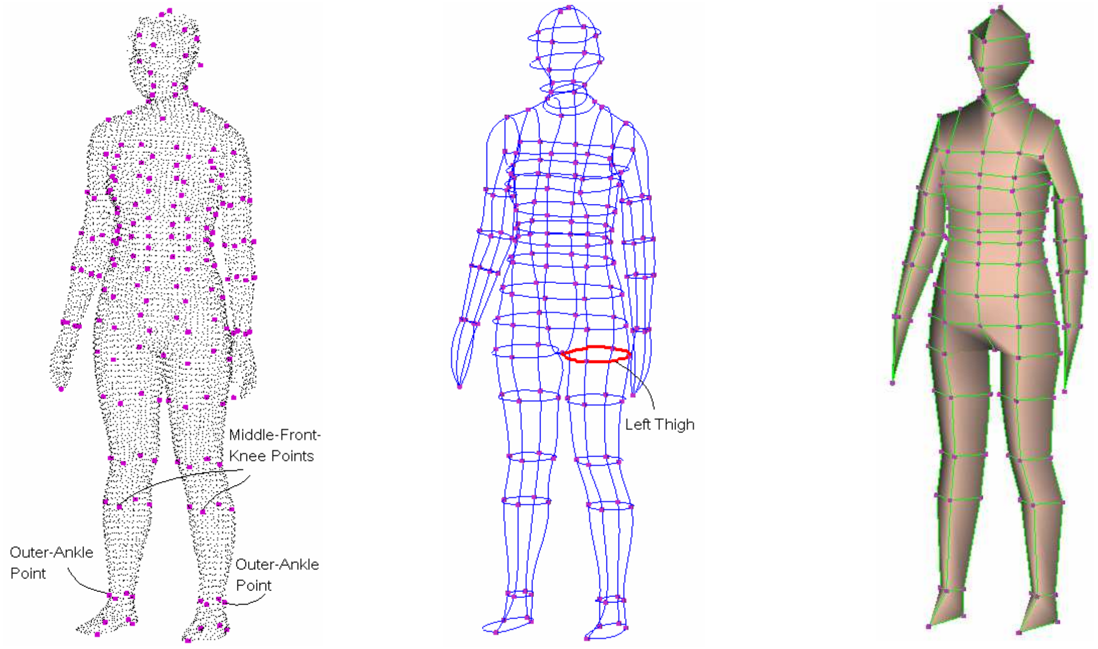
3.2 Semantic feature points

The location of the semantic feature points on the surface of a scanned human body can be roughly determined according to the anthropometrical rules by the key feature points. To accurate locate the semantic feature points, we should also adopt the feature extraction algorithm with fuzzy logic concept (ref. [1]). The basic idea is similar to the approach of detecting key feature points – using cutting planes and projection planes. The unnecessary detail of the procedure is omitted here. An example result of the key feature points and the semantic feature points extraction is as shown in Fig. 4a. Every semantic feature point has its anthropometrical meaning (e.g., the middle-front-knee points and the two outer-side-ankle points are specified in Fig. 4a), and the overall structure of the semantic feature points is fixed (e.g., a knee point cannot appear at the elbow region).

3.3 Wireframe structure

Now, to construct the feature wireframe, we need to link feature points with parametric curves. The linking curves are called feature curves as each curve has its semantic meaning according to sizing dimensions (e.g., four bold red curves specified in Fig. 4b give the final left thigh girth). The curves should pass through the feature points and approximate the shape of scanned human bodies. The parametric curves utilized in our implementation are 4th-order Bézier. Each curve has four control points. The first and the last control points are coincident with feature points, so only the middle two control points can be adjusted to approximate the scanned body shape. Every feature curve lies on a plane, which is determined by the anthropometrical rules (e.g., the plane to determine lower waist curves passes the bellybutton and parallels the ground). When computing the control points of a curve c_f , we first intersect the cloud points by the plane containing this curve to obtain a contour of points (e.g., the red points in Fig. 5b). Then, using the semantic feature information and the two endpoints of c_f to select points for approximating the feature curve (e.g., see the red points in Fig. 5c). Here, the semantic feature information is usually a bounding box to specify where is the possible region of c_f . Finally, a least-square fitting [24] is adopted to determine the positions of middle 2 control points. The procedure of computing an example feature curve – one segment of chest curve is shown in Fig. 5. After determining all feature curves, the feature wireframe is smoothed by adjusting the control points to let neighboring coplanar curves have G^1 continuity. As the feature curves are actually measurement curves for sizing dimensions, by the feature wireframe, the full dimension table of a scanned human can be easily calculated.

Since the feature wireframes are consistent to all human models, topology graphs by connecting feature curves with face entities are also consistent to all human models. This topology graph is a starting point of generating feature patches interpolating the feature wireframe. Fig. 4c shows an example topology graph according to the feature wireframe shown in Fig. 4b. It is easy to find that not only 4-sided patches but also 5-sided patches are included.

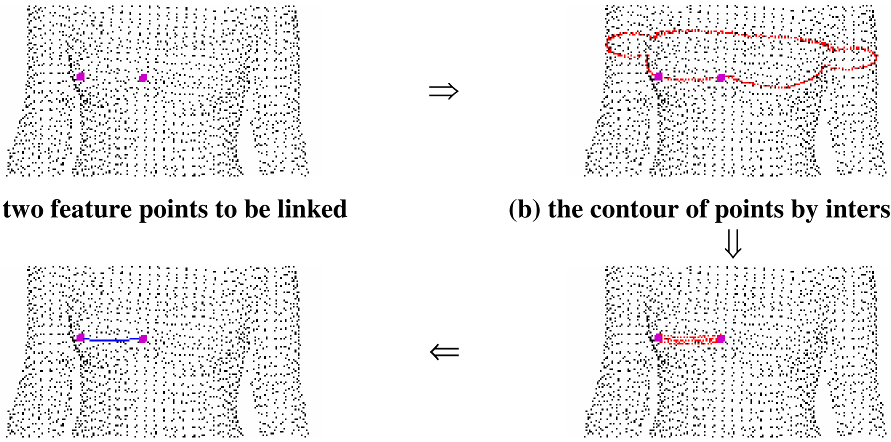


(a) with all feature points determined

(b) the final feature wireframe

(c) topology graph

Fig. 4 Feature wireframe and it related topology graph



(a) two feature points to be linked

(b) the contour of points by intersection

(d) resultant feature curve

(c) point to approximate the curve

Fig. 5 Steps of determining a feature curve

4 Feature patches Generation

As mentioned at the beginning of this paper, a feature human model without feature patches usually makes troubles to exactly locate cloth vertices when designing them around the body. For example, if the position of a vertex actually needs to be determined by points on a patch, encoding its position by the points on the boundaries of the patch may distort its final location. Also, without the in-between surface information, the shape of a human model is not well defined. Therefore, including feature patches is necessary to a parameterized feature human model. The generated feature patches should interpolate the feature wireframe constructed in section 3, and maintain enough details from the cloud points. In this section, the feature patches interpolating the feature curves are first generated using *Gregory patch* [25, 26]; the feature patches are then updated according to the scanned points by a voxel-based algorithm; finally, the mesh surface is adjusted to become symmetric.

4.1 Interpolation surface

We start generating the feature patches from the topology graph of a human body. By each face on the topology graph, we fill a Gregory patch which interpolates not only the feature curves, but also the cross tangents.

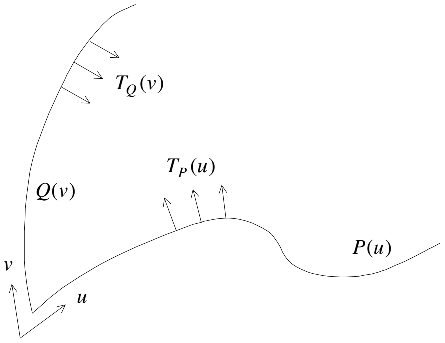


Fig. 6 Define a Gregory corner interpolator

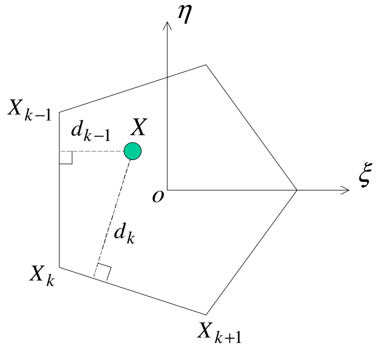


Fig. 7 P_G of a Gregory patch with five sides

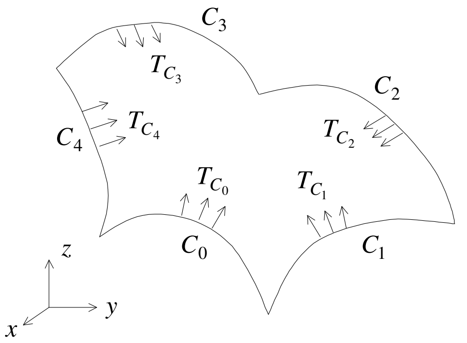


Fig. 8 Define a Gregory patch

Let $P(u)$: $0 \le u \le 1$ and $Q(v)$: $0 \le v \le 1$ be two regular curves in $\Re^3$ with $P(0) = Q(0)$, and $T_P(u) : 0 \le u \le 1$ and $T_Q(v) : 0 \le v \le 1$ be two C^1 vector functions in $\Re^3$ satisfying $T_P(0) = \left.\frac{dQ(v)}{dv}\right|_{v=0}$ and $T_Q(0) = \left.\frac{dP(u)}{du}\right|_{u=0}$, the *Gregory corner interpolator* of the four, { $P(u)$, $Q(v)$, $T_P(u)$, $T_Q(v)$ }, is a surface in $\Re^3$ defined by

$$r(u,v) = P(u) + vT_P(u) + Q(v) + uT_Q(v) - P(0) - vT_P(0) - uT_Q(0) - uv\frac{vT'_P(0) + uT'_Q(0)}{u+v} . \tag{1}$$

The Gregory corner interpolator function $r(u,v)$ agrees with $P(u)$ and $Q(v)$ along the two sides (i.e., $r(u,0) = P(u)$ and $r(0,v) = Q(v)$) (Fig. 6). Also, its partial derivatives with respect to *u* and *v* agree with $T_P(u)$ and $T_Q(v)$ along the respective sides – $\left.\frac{\partial r(u,v)}{\partial v}\right|_{v=0} = T_P(u)$ and $\left.\frac{\partial r(u,v)}{\partial u}\right|_{u=0} = T_Q(v)$ since $T_P(0) = Q'(0)$ and $T_Q(0) = P'(0)$. For an *n*-sided 3D surface, *n* such interpolator functions can be defined on the *n* corners; the final surface is the weighted sum of the *n* functions [25-28].

The parametric domain of a Gregory patch with *n* sides is defined as a unit length regular *n*-gon in the $\xi-\eta$ domain. We name the parametric domain of a Gregory patch G as P_G, where all corners X_k ($k = 0, 1, \ldots, n-1$) are ordered in the anti-clockwise (as shown in Fig. 7). Given a point $X = (\xi_0, \eta_0)$ inside P_G, when computing its three dimensional position defined by a Gregory corner interpolator $r_k(u_k, v_k)$, the parameters (u_k, v_k) of the point corresponding to the *k*th corner X_k are defined as

$$(u_k, v_k) = (\frac{d_{k-1}}{d_{k-1} + d_{k+1}}, \frac{d_k}{d_{k-2} + d_k}) \tag{2}$$

where d_k represents the perpendicular distance from X to the side $X_k X_{k+1}$. It is easy to find that if (ξ_0, η_0) lies on the side $X_k X_{k+1}$, $v_k = 0$ since $d_k = 0$; if (ξ_0, η_0) is on $X_{k-1} X_k$, $u_k = 0$ since $d_{k-1} = 0$; when (ξ_0, η_0) and X_{k+1} coincides, we have $u_k = 1$ by equation (2); and when (ξ_0, η_0) and X_{k-1} coincides, we have $v_k = 1$.

If $C_0(u)$, $C_1(u)$, …, $C_{n-1}(u)$ are *n* regular 3D curves that form a closed loop in three dimensional space, that is $C_k(1) = C_{(k+1) \bmod n}(0)$ ($k = 0, 1, \ldots, n-1$), and $T_{C_0}(u)$, $T_{C_1}(u)$, …, $T_{C_{n-1}}(u)$ are *n* continuous 3D vector functions defined on the $C_k(u)$ s respectively, the *Gregory patch* of $C_k(u)$ s and $T_{C_k}(u)$ s is defined as a mapping from P_G to $\Re^3$

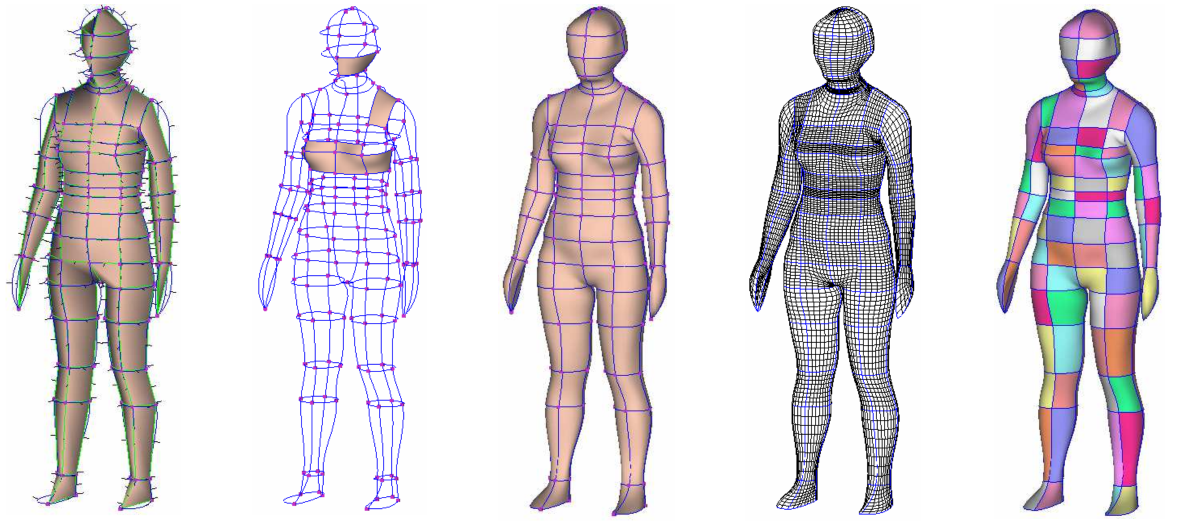
$$G(X) = \sum_{k=0}^{n-1} w_k(X) r_k(u_k(X), v_k(X)) \tag{3}$$

where $w_k(X) = \frac{\prod_{j \neq k-1, k} d_j^2}{\sum_{l=0}^{n-1} \prod_{j \neq l-1, l} d_j^2}$, and $r_k(u_k, v_k)$ represents the Gregory corner interpolator function for the *k*th corner of the four items { $C_k(u)$, $\overline{C}_k(v)$, $T_{C_k}(u)$, $\overline{T}_{C_k}(v)$ }, $\overline{C}_k(u) = C_k(1-u)$, and $\overline{T}_{C_k}(u) = T_{C_k}(1-u)$.

In our feature patch interpolation, the $C_i(u)$ s are the feature curves in a feature wireframe. What are the $T_{C_i}(u)$ s? After the topology graph Γ of a human model is determined, the cross normal N_i of the *i*th edge on Γ is computed by normalizing the mean of its left face normal and right face normal. The *i*th edge on Γ has a corresponding feature curve $C_i(u)$. As $C_i(u)$ is a parametric curve – 4th-order Bezier, its tangent vector function satisfies C^1. Thus, we determine $T_{C_i}(u)$ by

$$T_{C_i}(u) = N_i \times \frac{\partial C_i(u)}{\partial u}, \tag{4}$$

which also satisfies C^1 continuity. To have a continuous connecting across the feature curves, the Gregory patches on different sides of a feature curve should have the same number of grids along the curve. Fig. 9a shows the topology graph with cross normal vectors displayed as short black line segments; in Fig. 9b, some Gregory patches have been filled on the feature wireframe; Fig. 9c, 9d, and 9e give the result of all interpolation patches generated.



(a) topology graph with cross normals **(b) curves and patches on a human body** **(c) all feature patches generated** **(d) the mesh structure of the patches** **(e) checkerboard pattern to verify the feature patches**

Fig. 9 Gregory interpolation of feature patches

4.2 Surface refinement

From Fig. 9c, it is found that the interpolation surface does not give necessary detail of the human body's surface. To add the details, we introduce an algorithm to iteratively improve the fitting accuracy by minimizing the shape difference between the mesh surface *M* and the scanned data. To have a better performance at the computing speed, a voxel-based technique is adopted. The procedure of surface refinement consists of three basic steps: 1) voxel construction, 2) vertex position update, and 3) mesh relaxation. The second and the third steps are executed iteratively until a satisfied mesh is obtained.

Voxel Construction: The work in this step is to prepare voxels for updating the position of each vertex on the interpolation mesh surface. If the size of each voxel is chosen as δ (the criterion of choosing δ will be given in the description of the second step), and the bounding box of scanned cloud points is determined as $(x_{\min}, y_{\min}, z_{\min}) \times (x_{\max}, y_{\max}, z_{\max})$, we can construct $m \times n \times l$ voxels where $m = \frac{x_{\max} - x_{\min}}{\delta} + 1$, $n = \frac{y_{\max} - y_{\min}}{\delta} + 1$, and $l = \frac{z_{\max} - z_{\min}}{\delta} + 1$. Each voxel contains a list of points within a specific range of coordinates. For a voxel $V_{i,j,k}$, its range of point coordinates is

$$[i\delta + x_{\min}, (i+1)\delta + x_{\min}) \times [j\delta + y_{\min}, (j+1)\delta + y_{\min}) \times [k\delta + z_{\min}, (k+1)\delta + z_{\min}).$$

Thus, by the index of a voxel, we can easily figure out the points within a range in a constant computing time.

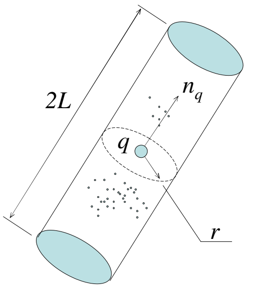


Fig. 10 Effective region Γ around a vertex

Vertex Position Update: For every vertex on the interpolation surface, we update its position by the points in a cylindrical region around it. As illustrated in Fig. 10, for a vertex *q* with its normal direction n_q on the mesh surface, all scanned points in the region Γ specified by *r* and *L* should be considered. Here we use a threshold of *r* = 1.125cm and *L* = 1.5 cm in our experiments. Among all the scanned points fall in Γ, we search out a point *p* that is farthest to *q*. The vertex *q* is moved to be coincident with *p*. To accelerate the point-selecting computing, the voxels constructed in the first step is utilized. If the coordinate of *q* is (x_q, y_q, z_q), only the points in the nine voxels $V_{i\in[\alpha-1,\alpha+1], j\in[\beta-1,\beta+1], k\in[\gamma-1,\gamma+1]}$ are tested, where

$$\alpha = INT(\frac{x_q - x_{\min}}{\delta}), \ \beta = INT(\frac{y_q - y_{\min}}{\delta}), \ \gamma = INT(\frac{z_q - z_{\min}}{\delta})$$

with *INT*(…) computing the truncation of a real number. Therefore, the chosen of δ is related to the value of *L*. When letting $\delta = L$, all possible scanned points falling in Γ have been include in the voxels whose indexes satisfy $i \in [\alpha - 1, \alpha + 1]$, $j \in [\beta - 1, \beta + 1]$, and $k \in [\gamma - 1, \gamma + 1]$.

Mesh Relaxation: Simply updating the position of each vertex on *M* by the above point selection procedure will not result in a very attractive mesh, because neighboring parts of *M* could get mapped to disparate parts on the scanned data. To overcome this disparity, the relative deformation between vertices within *M* is expected to be minimum. Also, the vertices derived from the feature curves should be constrained on the feature curves. Given the current position of a vertex *q*, the relaxation energy at *q* is defined as

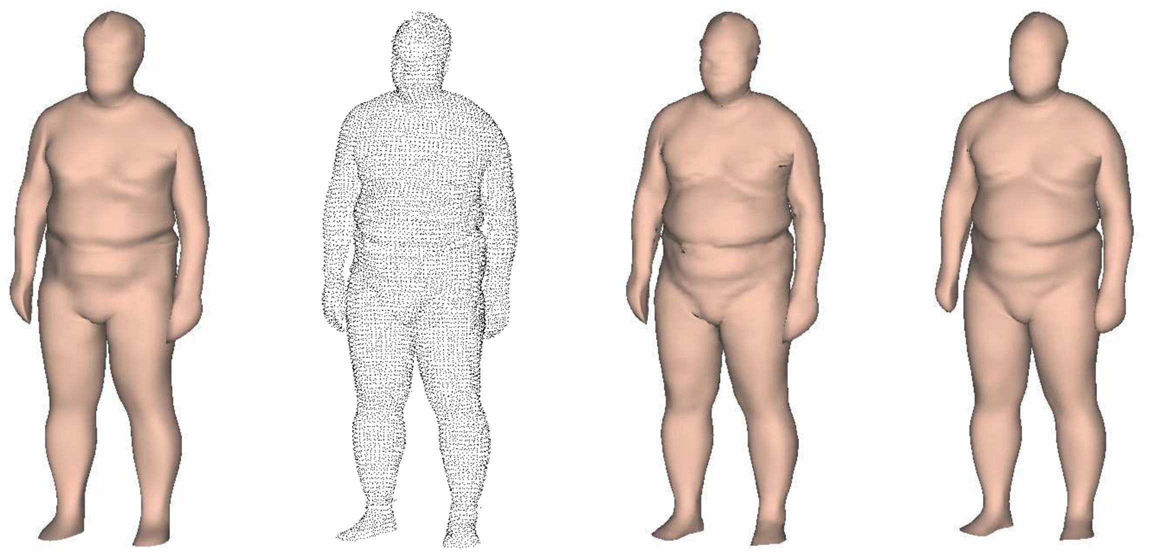
$$E = \sum_{j} (\|v_j q\| - l^0_{v_j q})^2 + E_{FC} \tag{5}$$

where v_j is a neighboring vertex of *q* on *M*, $l^0_{v_j q}$ is the distance between v_j and *q* on *M* before the surface refinement, and E_{FC} is the energy defined on the vertices lying on a feature curve. In a polygonal mesh, not only the vertices with edges connecting *q* but also other vertices on the faces containing *q* without edge connecting to *q*, are called its neighborhood – so when computing *E*, they also need to be counted. For *q* originally lies on a feature curve, we have

$$E_{FC} = (q - q_0)^2 \tag{6}$$

where q_0 is the position of *q* on the feature wireframe; otherwise, $E_{FC} = 0$. The mesh relaxation is actually the procedure of minimizing the energy *E* vertex by vertex.

The mesh refinement is performed by iteratively running step 2 and step 3 until the average distance error between the vertices on *M* and the scanned points is less than a threshold ε (our implementation adopts ε =0.1mm). However, the refinement result is annoyed by the high frequency noises in the scanned cloud points – see Fig. 11c, the noises effect the smoothness of resultant surface a lot. We simply solve this problem by applying the 2nd-order fairing operator [29] on each vertex one time after every ten iterations of step 2 and step 3. The final refinement result with all the factors considered is much better (e.g., Fig.11d). Benefit by the semantic feature technology, the surface construction can be finished within one minute on a computer with standard settings (PIII 900 MHz CPU + 256 MB RAM), where the scanned point cloud has about 100,000 points.



(a) interpolation surface **(b) scanned data** **(c) with noises** **(d) result with $\varepsilon = 0.1$**

Fig. 11 Surface refinement

4.3 Making symmetric

The human body generated from the scanned data is usually asymmetric. When the human model is generated for sizing survey or for mannequin manufacturing, the symmetric models are requested. If this is the case, the refined mesh surface has to be further modified. Let us name the resultant surface after mesh refinement as M^+, the feature wireframe as F^+, and the scanned cloud points as S^+. First of all, the entire data set of a human body including M^+, F^+, and S^+ are transformed to let the crotch point on the origin. For every feature curve C in F^+, there is a dual curve C^* in F^+, where they should be symmetric on a symmetric wireframe. Also, for each feature node P in F^+, there should exist a symmetric dual node P^* of it in F^+. By this property, we can construct a feature wireframe F^- which is symmetric to F^+. In detail, for each feature node P, we update its position by the mirrored coordinate of P^* according to the x-z plane, and vice-versa; for each feature curve C, its control points are updated by the mirrored copy of the data points on C^*. Then, we mirror the coordinates of all points in S^+ to obtain S^-. After applying the surface interpolation and refinement algorithms on F^- and S^-, we obtain a refined mesh surface M^- of the mirrored human body. The symmetric model surface is obtained by the interpolation of M^+ and M^-: $M^S = \frac{M^+ + M^-}{2}$; at the same time, we can have the symmetric feature wireframe F^S by $F^S = \frac{F^+ + F^-}{2}$. Fig. 12 shows a set of S^+, F^+, M^+, S^-, F^-, M^-, M^S and F^S.

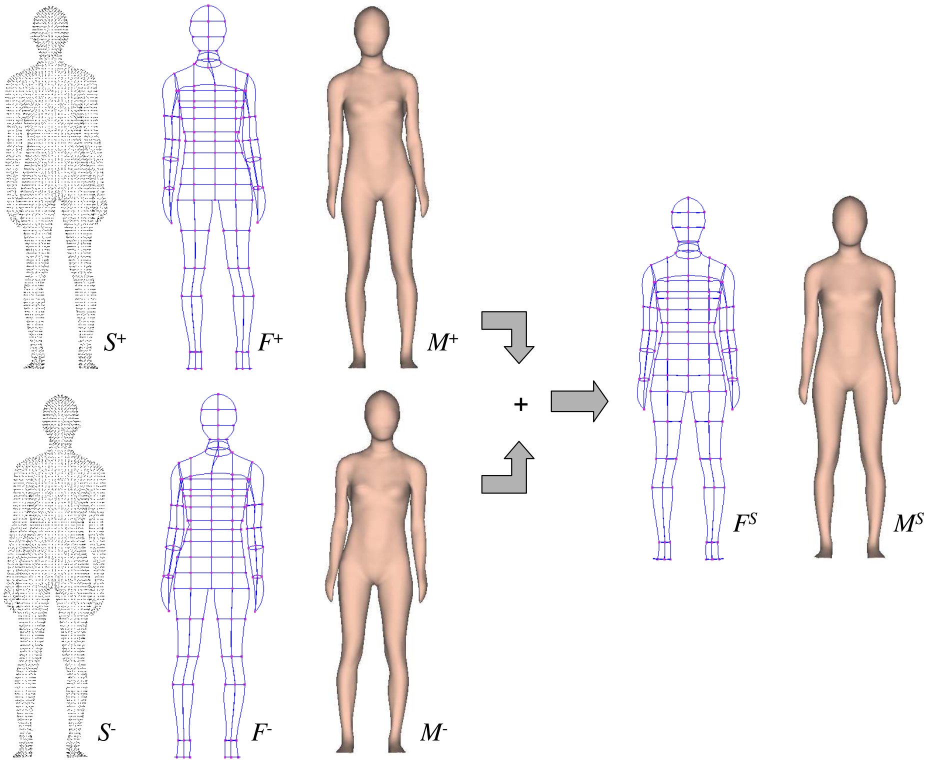


Fig. 12 Making symmetry

5 Parametric Design of Human Model

All the parameterized human bodies are stored in a 3D digital human model database Π. When we have enough data, a new human body can be generated according to the specified sizing dimensions by synthesizing example models from Π. Giving different sizing parameters, different human bodies can be synthesized – so called the parametric design of human bodies. One might ask why not just modify the sizing parameters related curves in a feature wireframe F to satisfy the given dimensions input, and then shift the displacement map of the human model mesh onto the modified wireframe. The reason is that such an approach hardly guarantees realistic shape in the resultant geometry. To guarantee the realistic, an example-based synthesis approach is adopted. In this section, the numerical optimization based synthesis algorithm is first introduced, and followed by the strategy of choosing examples for the synthesis.

5.1 Example model based synthesis

A smooth interpolation is sought to transform the sizing dimensions into a model in the body geometric space by using scattered examples as interplants. Many example-based methods [10-13] adopt mathematical interpolation functions to describe the interpolation; however, no matter how accurate an analytical interpolation function is, the approximation errors are not controlled when the given parametric vector has no coincident example in the sample set. Therefore, in order to minimize the approximation errors, we adopt a numerical optimization based scheme here.

Assume that we have *n* human bodies: H_i (*i* = 0, …, *n*-1), the synthesized human body H_s can be obtained through interpolation as $H_S = \mathrm{I}(w_0, H_0, w_1, H_1, \cdots, w_{n-1}, H_{n-1})$ where $\mathrm{I}(\cdots)$ is the interpolation function, w_is are the weights of interpolation satisfying $\sum_{i=0}^{n-1} w_i = 1$ and $w_i \geq 0$. We also define a measurement function $\Psi_M(H)$ whose input is a human body *H*, and output is a dimension vector of *H*. Forming a vector *D* by the input sizing parameters, $\|\Psi_M(H) - D\|$ gives the difference between the given human body *H* and the specified dimensions of parametric design. Also, in order to increase the degree-of-freedom, every human example H_i can be scaled by a scale factor α_i during the interpolation ($\alpha_i \geq 0$). Therefore, based on a set of example human bodies: H_is, the problem we are going to solve here is to determine the weights of interpolation and the scale factors.

Setting a vector *X* containing *2n* scales to be determined: $X = (\omega_0, \omega_1, \cdots, \omega_{n-1}, s_0, s_1, \cdots, s_{n-1})$, the synthesis human body on *X* is determined by $H_S = \mathrm{I}(\frac{\omega_0^2}{\sum_i \omega_i^2}, s_0^2 H_0, \frac{\omega_1^2}{\sum_i \omega_i^2}, s_1^2 H_1, \cdots, \frac{\omega_{n-1}^2}{\sum_i \omega_i^2}, s_{n-1}^2 H_{n-1})$ where $s_i^2 = \alpha_i$ and $\frac{\omega_j^2}{\sum_i \omega_i^2} = w_j$. Thus, when $\sum_i \omega_i^2 \neq 0$, the requirements on $w_i \geq 0$ and $\alpha_i \geq 0$ are satisfied by any *X*. The vector *X* is the variable to be determined during the optimization. The synthesis human body can be considered as a function of *X* – $H_s(X)$. The parametric design is formulated as an optimization problem, where we search for a configuration of *X* leads to the minimum difference between H_s and the inputs:

$$\boldsymbol{min}\ \|\Psi_M(H_s(X)) - D\|^2 \tag{7}$$

Based on the above equation, $J[X] = \|\Psi_M(H_s(X)) - D\|^2$ is defined as the objective function of numerical optimization. At the beginning of optimization, the values of *X* are given as $\omega_i = s_i = 1$. To ensure $\sum_i \omega_i^2 \neq 0$, we just simply fix the value of ω_0 when all $\omega_i^2 < 10^{-6}$ during the optimization.

We compute the optimized *J* with respect to *X* by a conjugate gradient method which includes the iterative process of computing gradients at current *X* and searching an optimum point along the conjugate direction [30]. The gradients are computed numerically. The unnecessary details of the conjugate gradient method are omitted here. During the iteration of optimization, the value of objective function decreases while the number of iteration increases. Usually, these two factors are utilized together to give the terminal criterion of iterations. Thus, our terminal condition is either $\frac{\left\| J[X^{i}] - J[X^{i-1}] \right\|}{J[X^{0}]} < \varepsilon$ or the iteration number is greater than $N_{\max}$, where $J[X^{i}]$ is the value of the objective function in the *i*th iteration (current value), $J[X^{0}]$ is the value of the objective function before optimization, $N_{\max}$ is the maximum iteration number, and ε is a small number (we choose $\varepsilon = 0.01\%$ and $N_{\max} = 100$ in all of our testing examples). Since the $\Psi_M(\cdots)$ function only has relationship to the feature wireframe of a human body, to speed up the numerical optimization, the interpolation is only applied to the feature wireframe but not the mesh surface of a human body during optimization. After determining the interpolation weights and the scale weights, the mesh surface of the new human model is generated at the final step.

5.2 Example choosing strategy

To support the numerical optimization based synthesis algorithm, appropriate examples are required to be performed as interpolants. The *appropriate* here means both the appropriate number and the models with appropriate sizing dimensions. Our example choosing strategy gives constraints on both of them.

Let's consider the number of examples first. If $\Psi_M(H_s) = \sum_i w_i \Psi_M(H_i)$, by the theorem of linear algebraic, for *m* input sizing parameters, we need as least *m* human models whose returned vectors of $\Psi_M(\cdots)$ are not linear correlated. However, we cannot guarantee $\Psi_M(H_s) = \sum_i w_i \Psi_M(H_i)$ as the measurements may be nonlinear to the synthesis; so the number of examples can only be determined by experiences. Too many examples lead to long computing time for the synthesis algorithm, while too small amount of examples can rarely get a human model fully satisfying the given dimensions (can only give an approximation). From our testing experience, using *4m* models with appropriate dimensions usually works well.

For a vector of input parameters *D*, the distance from a human model *H* to *D* in the sizing parameter space is defined as the Euclidean norm of $\Psi_M(H) - D$ as

$$L = \left\| \Psi_M(H) - D \right\|. \tag{8}$$

In our example selecting strategy, the smaller *L* is given from a human model *H*, the more appropriate the human body is to be served as a synthesizing example.

In summary, by a given *D* with *m* components, we sort all model in the 3D digital human model database Π according to their *L*s in ascending order. Then, the first *4m* models are selected as examples for the synthesis algorithm. If the optimization gives inaccurate result according to D (i.e., the iteration stops at the $N_{\max}$ criterion and the returned value of $J[X]$ exceeds some threshold), add m more models as the examples. The algorithm repeatedly selects models, and synthesizes models until the new model with accurate dimensions to D is obtained. The same as other example-based approaches, this approach also relies on the number of models store in Π. Thus, building a 3D digital human model database with a large number of models is quite an important work. To let the synthesized human model have the morphology features of geography, the human models born in different regions should be stored separately.

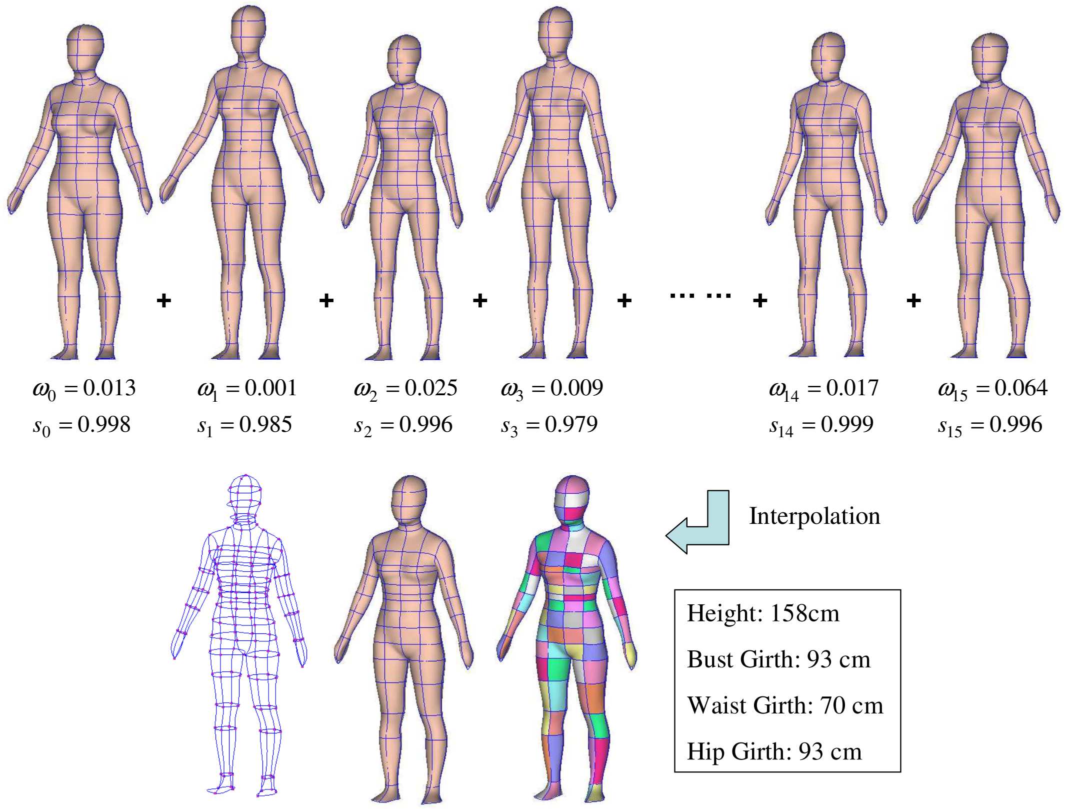


Fig. 13 Synthesis of a human body

5.3 Synthesis results

With the input sizing parameters, we can construct a lot of different human bodies. Fig. 13 shows an example female model generated by inputting sizing parameters (the parameters are listed in the figure); the utilized example models and their final weights and scales for synthesis are also depicted. Fig. 14 shows a serial of female models according to different parameters listed in Table 1. In Fig. 15, six male models are generated according to the dimensions given in Table 2. Fig. 16 shows the parametric design results of the female models with the same height but different hip sizes, and Fig. 17 gives the resultant male models through specifying the same height but different waist girths.

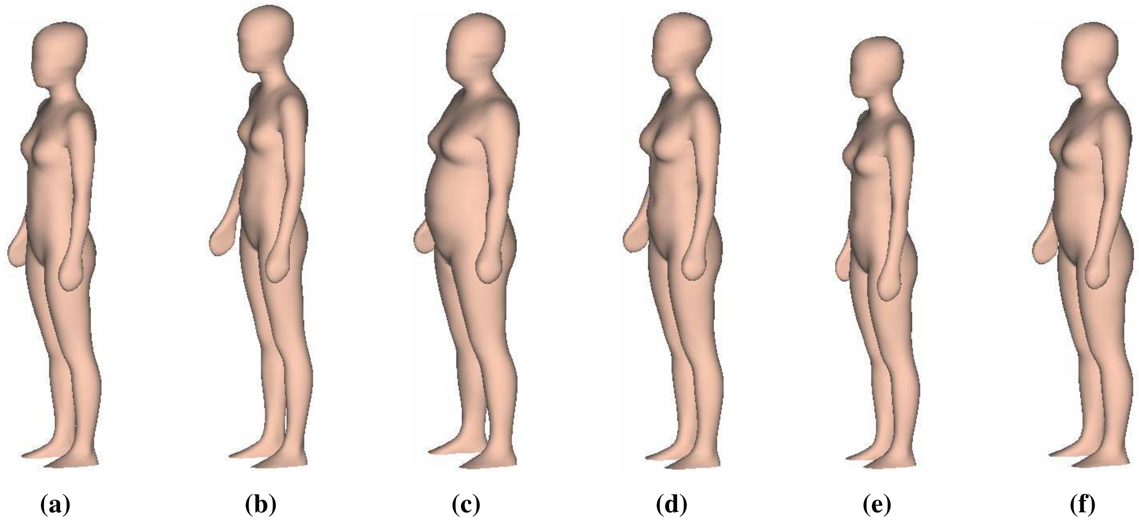


Fig. 14 Female models generated according to the dimensions listed in Table 1

Table 1 Input sizing parameters to generate female models in Fig. 14

Model	Height	Neck Girth	Bust Girth	Under Bust	Waist Girth	Hip Girth	Inseam-Length
(a)	159	32	85	74	66	89	69
(b)	168	32	83	77	66	90	77
(c)	165	39	107	94	92	108	70
(d)	163	33	92	77	72	95	70
(e)	156	31	85	73	65	87	67
(f)	159	34	95	82	75	96	68

Table 2 Input sizing parameters to generate male models in Fig. 15

Model	Height	Neck Girth	Chest Girth	Waist Girth	Hip Girth	Inseam-Length
(a)	175	42	110	105	109	74
(b)	184	39	97	84	101	84
(c)	169	39	95	86	98	70
(d)	175	40	98	85	99	76
(e)	170	40	100	90	99	69
(f)	174	45	115	112	114	71

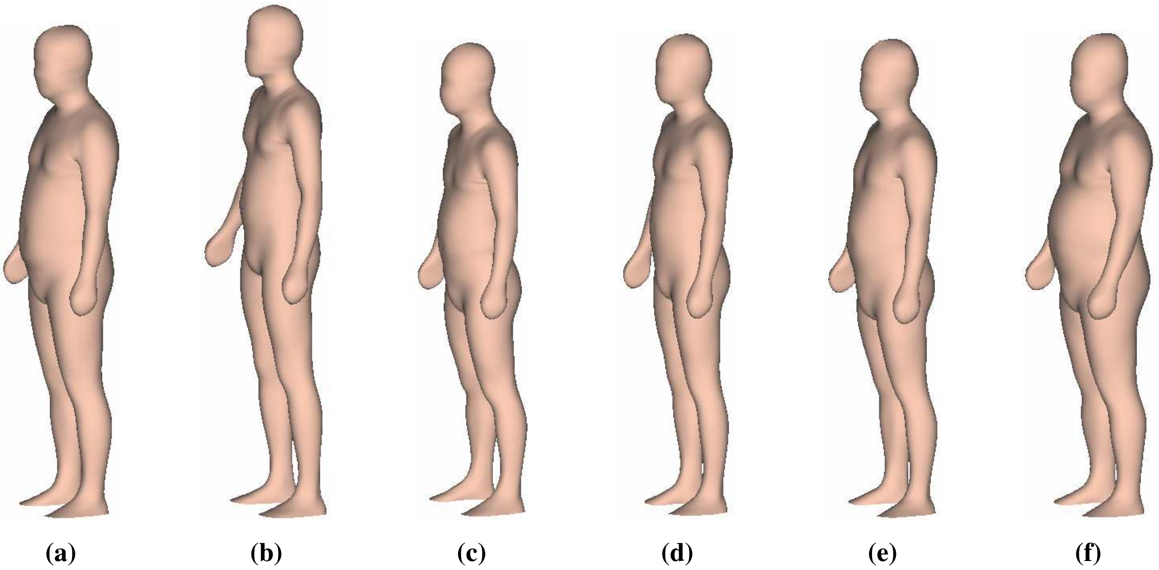


Fig. 15 Male models generated according to the dimensions listed in Table 2

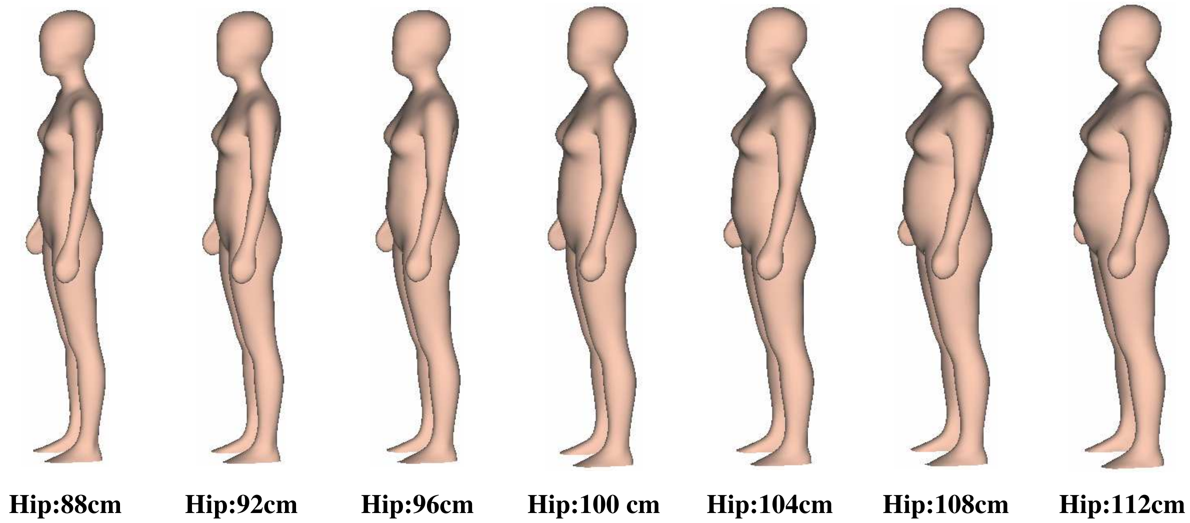


Fig. 16 With hip/height ratio changed (Height:165cm)

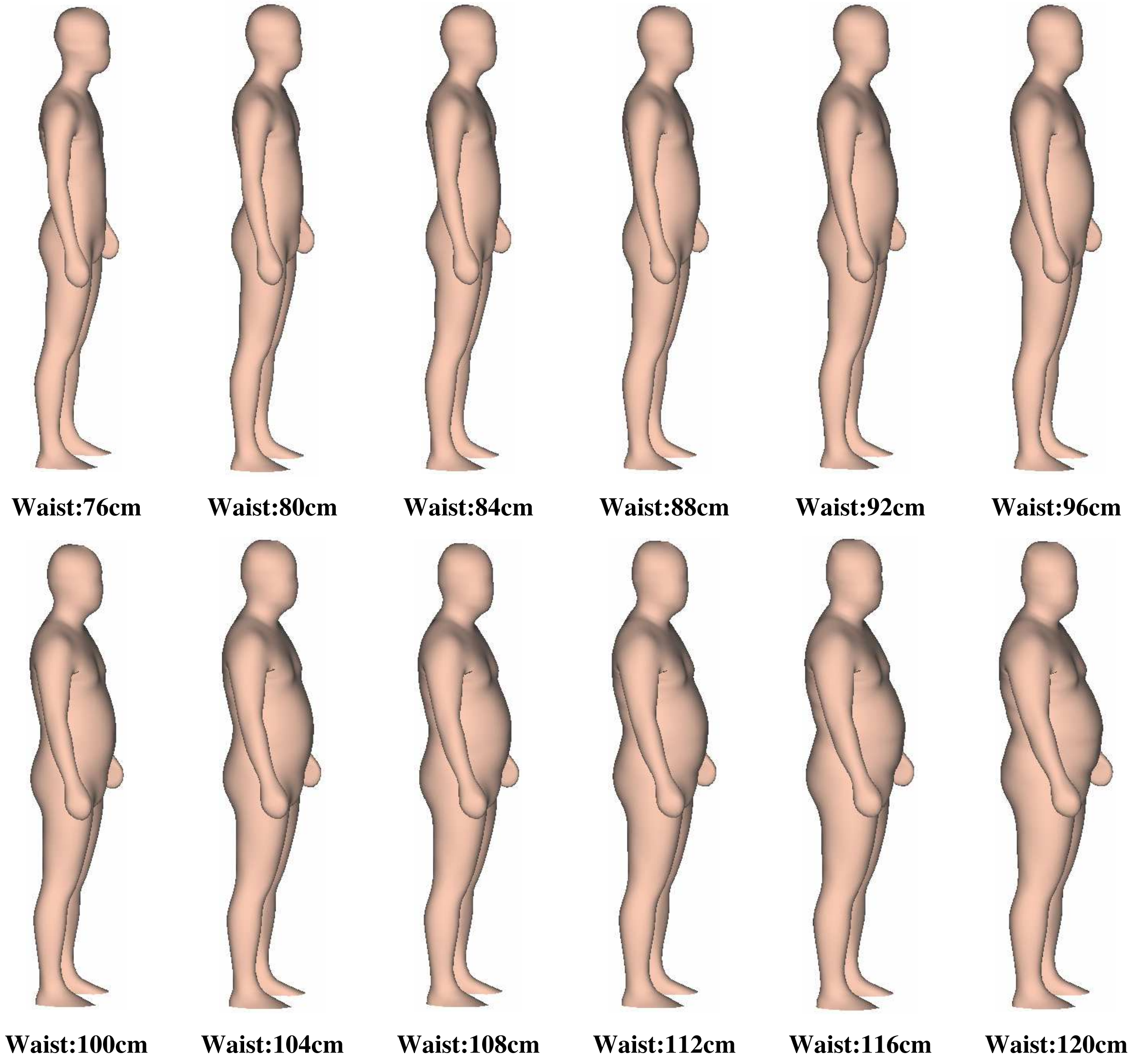


Fig. 17 With waist/height ratio changed (Height:175cm)

6 Application for Design Automation of Customized Clothes

The application, which gains the greatest benefit from the technique presented in this paper, is the design automation of customized clothes. The most significant differences of the human model generated in this paper and the feature human in [1] are the feature patches. The human model generated in [1] only have feature curves but not patches, so the clothes generated around the human model can only be encoded on feature curves or feature nodes – which greatly limits the freedom of design. When having the feature patches provided in this paper, not only structure curves which are the entities encoded in [3], but also the entire mesh surface of a piece of cloth is encoded around the feature human body. Fig. 18 gives a preliminary example of the design automation of customized apparel products. After designing and encoding a piece of waistcoat and a piece of pants on a parameterized human body *H* which guarantees fit, the encoded relationship between the clothes and the mannequin can be applied to different human models to generate the customized 3D clothes fitting individual bodies – so the design automation of customized apparel products is implemented. The H_is are from the synthesis results previously shown in Fig. 14.

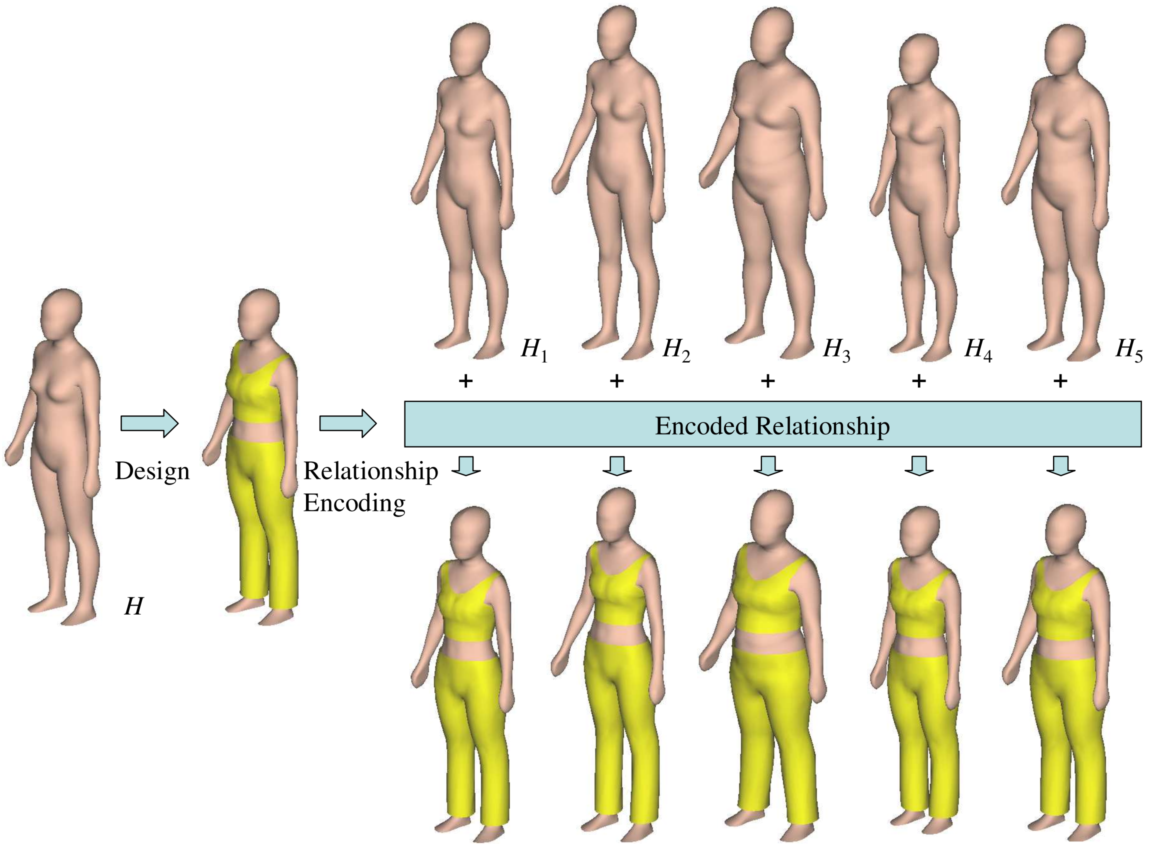


Fig. 18 Example of cloth design automation benefited from the technique of parameterized mannequins

7 Conclusion and Discussion

A new framework for generating feature-base whole human bodies according to the specified measurement dimensions is presented. Our contributions include: 1) the feature based parameterization approach of human bodies from the unorganized scanning points, which is more efficient than the template fitting approaches; 2) the numerical optimization based example synthesis method by giving sizing dimensions, where the approximation error is minimized. The constructed and synthesized feature models by our approach are patch-based, so not only feature nodes and curves but also feature patches are modeled. These feature entities give great benefits to the successive design automation of customized clothes. An example application for showing this benefit is also included in the paper.

There is one hidden assumption during the feature point extraction – though not necessarily identical, the posture of the scanned human body is similar. If the posture is quite different, the feature extraction method may give incorrect resultant points that will lead the failure of our parameterization approach. In this case, we need to specify the key feature points on the scanning points manually. A possible future work is to employ a more robust feature extraction algorithm for the key feature points.

Our parametric design algorithm is example-based, so if the specified dimensions are out of range (e.g., the hip to height ratio is much greater than all examples stored), the resultant synthesized human body will just be an approximation of the given sizing parameters; in other words, cannot achieve the exact given dimensions. If this is the case, one possible future work is to further modify the feature curves to satisfy the given dimensions, and then shift the displacement map of the synthesized model to fit the modified feature wireframe. The drawback is that the resultant model might be less realistic since it has been manually modified.

Finally, another research possibility is related to the input. Current input of the human construction approach presented is limited to the sizing dimensions. Photo is another convenience input for constructing a three-dimensional model of a human body. Therefore, we are going to borrow the idea from [2, 7] to develop a new method of human model construction by this input.

References

[1] Wang C.C.L., Cheng 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.

[2] Wang C.C.L., Wang Y., Cheng 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.

[3] 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.

[4] Scheepers F., Parent R.E., Carlson W.E., and May S.F., Anatomy-based modeling of the human musculature, Computer Graphics Proceedings, SIGGRAPH 97. ACM. 1997, pp.163-172. New York, NY, USA.

[5] Wilhelms J., and Van Gelder A., Anatomically based modeling, Computer Graphics Proceedings, SIGGRAPH 97. ACM. 1997, pp.173-180. New York, NY, USA.

[6] Dekker L., 3D human body modeling from range data, Ph.D. Thesis, University College London, 2000.

[7] Hilton A., Beresford D., Gentils T., Smith R., Sun W., and Illingworth J., Whole-body modelling of people from multiview images to populate virtual worlds, Visual Computer, vol.16, no.7, 2000, pp.411-436.

[8] Lee W.S., Gu J., and Magnenat-Thalmann N., Generating animatable 3D virtual humans from photographs, Computer Graphics Forum, vol.19, no.3, 2000, pp.1-10.

[9] Allen B., Curless B., and Popović Z, The space of human body shapes: reconstruction and parameterization from range scans, ACM Transactions on Graphics, vol.22, no.3, pp.587-594.

[10] Seo H., and Magnenat-Thalmann N., An Example-Based Approach to Human Body Manipulation, Graphical Models, vol. 66, no.1, pp.1-23, 2004.

[11] Blanz V., and Vetter T., A morphable model for the synthesis of 3D faces, Computer Graphics Proceedings. SIGGRAPH 99. ACM. 1999, pp.187-194. New York, NY, USA.

[12] Sloan P.P.J., Rose C.F., and Cohen M.F., Shape by example, Proceedings of the 2001 symposium on Interactive 3D graphics, pp.135-143, 2001.

[13] Lewis J.P., Cordner M., and Fong N., Pose space deformation: a unified approach to shape interpolation and skeleton-driven deformation, Computer Graphics Proceedings. SIGGRAPH 2000. Conference Proceedings. ACM. 2000, pp.165-172. New York, NY, USA.

[14] Ma W., and He P., B-spline surface local updating with unorganised points. Computer Aided Design, 1998, 30(11): 853-862.

[15] Ma W., and Zhao N., Smooth multiple B-spline surface fitting with Catmull-Clark subdivision surfaces for extraordinary corner patches, Visual Computer, vol.18, no.7, Nov. 2002, pp.415-36.

[16] Barhak J., and Fischer A., Parameterization for reconstruction of 3D freeform objects from laser-scanned data based on a PDE method. The Visual Computer, vol. 17, no. 6, 2001, pp.353-369.

[17] Sienz J., Szarvasy I., Hinton E., and Andrade M.L., Computational modelling of 3D objects by using fitting techniques and subsequent mesh generation. Computers & Structures, vol.78, no.1-3, 2000, pp.397-413. Publisher: Elsevier, UK.

[18] Au C.K., and Yuen M.M.F., Feature-based reserve engineering of mannequin for garment design. Computer Aided Design, 1999, 31(12): 751-759.

[19] Shah J.J., and Mantyla M., *Parametric and Feature-based CAD/CAM: Concepts, Techniques, and Applications*. John Wiley & Sons, New York, NY, 1995.

[20] Li C.L., and Hui K.C., Feature recognition by template matching, Computers & Graphics, vol.24, no.4, Aug. 2000, pp.569-82.

[21] Vergeest J.S.M., Horvath I., and Spanjaard S., Parameterization of freeform features, Proceedings International Conference on Shape Modeling and Applications. IEEE Computer Society. 2001, pp.20-9. Los Alamitos, CA, USA.

[22] van den Berg E., Bronsvoort W.F., Vergeest J.S.M., Freeform feature modelling: concepts and prospects, Computers in Industry, vol.49, no.2, Oct. 2002, pp.217-33.

[23] Vergeest J.S.M., Spanjaard S., and Song Y., Directed mean Hausdorff distance of parameterized freeform shapes in 3D, The Visual Computer, vol.19, no.1, pp.1-13, 2003.

[24] Mortenson M.E., *Geometric Modeling (2nd Edition)*. Wiley: New York, 1997.

[25] Gregory J.A., N-sided surface patches, Mathematics of Surfaces. Proceedings of a Conference. Clarendon Press. 1986, pp.217-32. Oxford, UK.

[26] Gregory J.A., and Yuen P.K., An arbitrary mesh network scheme using rational splines, Mathematical Methods in Computer Aided Geometric Design II, Lyche T. and Schumaker L.L. (eds.), pp.321-329, 1992, Academic Press, Inc.

[27] Hall R., and Mullineux G., Shape modification of Gregory patches, The Mathematics of Surfaces VII, Goodman T. and Martin R. (eds.), pp.393-408, 1997, Information Geometers.

[28] Mullineux G., Improvement of free-form surfaces for product styling applications, Computer-Aided Design, vol.34, no.12, pp.871-880, 2002.

[29] Kobbelt L., Campagna S., Vorsatz J., and Seidel H.P., Interactive multi-resolution modeling on arbitrary meshes, SIGGRAPH 98 Conference Proceedings. ACM., 1998, pp.105-114. New York, NY, USA

[30] Belegundu A.D., and Chandrupatla T.R., *Optimization Concepts and Applications in Engineering*, Upper Saddle River, N.J.: Prentice Hall, 1999.