Automatic Human Body Reconstruction Based on Orthogonal Image Contour

Zhong Yueqi[1,2]
(1. Key Laboratory of Textile Science & Technology, Ministry of Education, Donghua University, ShangHai 201620;
2. College of Textiles, Donghua University,ShangHai 201620)

Abstract: With the development of online apparel retailing, the shape of the human body becomes the keystone when evaluating the dressing result in the cyberspace. In this paper, an automatic method has been presented to generate an avatar by deforming a candidate model against two orthogonal images of the online shopper. The candidate model is prepared from range scanned data. Mean value deformation is initiated to make it coincide with the input image contours. The correspondence between control points on the candidate model and points on the image contour are built automatically by applying positional and directional constraints respectively. Using the proposed method, we can approximate the body shape of the online shopper effectively by preserving his/her features.

Keywords: Computer graphics;3D human body;2D orthogonal image contours;Mean value geometry deformation

0 Introduction

Browsing the 3D dressing result from internet requires the fidelity in both visual effects and fit/ease evaluation, since the garment can not be physically tried-on in the cyberspace. The traditional anthropometric size chart without the detailed shape variations can not provide the desired satisfaction for realism. An ideal approach is to represent the customer with an avatar that has the same body shape, with which the dressing style and fit/ease information can be illustrated.

3D shape morphing has been investigated for years. The common technique for deforming articulated characters is to define the position of the surface geometry as a function of an underlying skeletal structure or set of control parameters. This method usually assumes that the point displacements are generated by a weighted set of (usually linear) influences from neighboring joints. Recent advances in this area can be found in the work of Lewis *et al.* [1], Singh and Kokkevis [2], and Sloan *et al.* [3]. In the work of Capell [4], the character was embedded in a coarse volumetric control lattice, which provided the structure needed to apply the finite element method. Line constraints along the bones of simple skeletons were introduced to incorporate skeletal controls.

A general approach for the whole body reconstruction is to parameterize the models over a common base mesh [5-7]. This approach splits the meshes into matching patches with an identical inter-patch connectivity. Each set of matching patches is parameterized on a common convex planar domain. One advantage of this approach is that it naturally supports feature correspondence by using feature vertices as corners of the matching patches. The main challenge in mapping the models to a single base mesh is to construct identical inter-patch connectivities. The vast majority of the methods use heuristic techniques that work only when the models have nearly identical shape.

In the work of Allen *et al.* [8, 9], an example-based method was presented for calculating

Foundations: Specialized Research Fund for the Doctoral Program of Higher Education (SRFDP)(No. 20070255003);
Brief author introduction: Zhong Yueqi(1972-), male, Associate Prof., mainly engaged in the area of virtual garment simulation, virtual human body reconstruction and digitalized textiles/apparels. E-mail: phdzyq@gmail.com

skeleton-driven body deformations. The variability of human shape was captured by performing principal component analysis (PCA) over the displacements of the template points. A related displacement-mapped technique, without hole-filling, was also developed by Hilton *et al.* [10].

Pauly *et al.* [11] reported that the template models could be retrieved from the database to conform to the input data. Kraevoy *et al.* [12] proposed a cross-parameterization method to compute a low-distortion bijective mapping between models that satisfied the user prescribed constraints. Anguelov *et al.* [13] introduced a data-driven method for building a human shape model that spans variation in both subject shape and pose based on a representation that incorporates both articulated and non-rigid deformations. Mohr *et al.*[14] extended the linear blend skinning by adding a relatively small number of joints that were simply related to the original skeletal parameters to avoid the linearly blended matrix became degenerate.

The reconstruction of 3D objects from 2D images has long been of interest to researchers in computer vision and computer graphics. Our objective in this paper is to reconstruct a smooth 3D model from two orthogonal images, and in particular to compute a 3D model which projects exactly to the contour in the image. Transferring a known 3D model to a given 2D shape falls into the research region of “sketch-based deformation”, which is heavily investigated in recent years. After sketch recognition, the core is to determine the corresponding mesh vertices and their transformed positions respectively. In the work of Zimmermann *et al.* [15], an automatic correspondence between the reference path on the model and the sketched strokes was built using bounding volumes. Laplacian Surface Editing [16] was applied on the user defined silhouettes to manipulate the deformation. Prasad *et al.* [17] reconstructed the 3D counterpart from the apparent contour by minimizing a surface smoothness objective function. The correspondence was manually selected by the user. Yang *et al.* [18] proposed a sketch recognition algorithm by applying a set of given 2D templates to the sketch. Once a best template was found, a 3D object was constructed using a series of measurements that were extracted from the labeled 2D sketch. In the system built by Kho and Garland [19], the user drawn curve was regarded as a reference curve on the image plane, which defined the part of the surface under deformation. The sketch-based deformation was achieved via setting the correspondence between the target curve and the reference curve in terms of linear interpolation.

Throughout the literature, the most related research is the one reported by Kraevoy *et al.* [20]. They used mean value deformation to match the template models with the drawn contours. The finding of correspondence was regarded as an optimization problem using a hidden Markov model.

1 Automatic skeleton generation for image contours

Based on our proposed methodology, the online shopper is required to provide two orthogonal images under a formatted pose with single colored background, and the arm portion of the side view should be manually separated, as shown in Fig.1a. Canny edge detection is then performed to extract silhouette of the body shape (Fig. 1b) since the image is well-characterized. After which, the orthogonal images will be mounted to extract the landmarks and skeleton, as shown in Fig.1c.

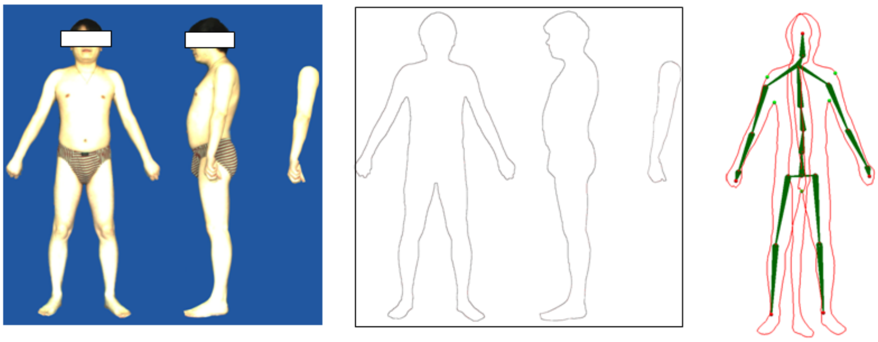

Fig.1 Pre-processing the image contours (from left to right): a) Original image. b) Original image contours. c) Mounted contours with landmarks and skeleton.

Denote the side-view contour as C_s, the side-view arm contour as C_{arm}, and the front-view contour as C_f, and note that C_s, C_{arm} and C_f are all located in XY plane at this point, the pre-processing procedure of landmark location and skeleton extraction follows.

Step 1. Given n points on C_s, find out the axis of C_s (denoted as C_{sAxis}) by connecting the mid-point of a series of cutting lines, as shown in Fig.2. For ith point p_i, a horizontal segment can be formed to find out the intersection point with C_s, denoted as q_i. After computing the mid-point of $p_i q_i$ (i=1.2,…n) and then sorting the set of mid-points in Y direction, the axis C_{sAxis} can be obtained accordingly.

Step 2. Find out the highest points on the arm contour C_{arm} as $p_{shoulder}$, form a horizontal line from $p_{shoulder}$ to get the intersection point with C_{sAxis} as $p'_{shoulder}$, and then move C_{arm} from $p_{shoulder}$ to $p'_{shoulder}$ to mount the arm contour.

Step 3. Find out the crotch point with the following algorithm:

From C_f, set the target zone as $x \in [-0.1h, 0.1h]$ and $y \in [-h, h]$, where h *equals to the length of the head.

* From the experience of anthropometry, the average adult human figure is about 7 ~8 heads tall [23]. In this paper, we choose 7 as the benchmark length. Although the automatic body measurement depends on the target zone defined by the head length, the range is loosely defined. Hence the head length will not jeopardize the final result, as proved in our previous work [21].

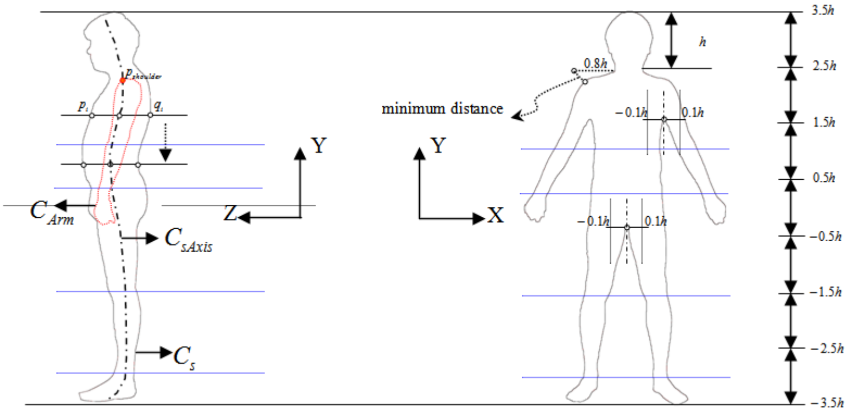

Fig. 2. Illustration of automatic image contour measurement

Find out the point with highest y-coordinate in this zone as the crotch point p_{crotch}, then form a horizontal line from p_{crotch} in YZ plane, intersect with C_{sAxis} at a point p'_{crotch}. Now rotate the side view contours (C_s, C_{sAxis} and C_{arm})from XY plane to YZ plane, and then make p_{crotch} and p'_{crotch} coinciding with each other. The z value of p'_{crotch} in YZ plane will be taken as the z value of p_{crotch} in 3D space, as shown in Fig.2. From this moment, we will always calculate the x and y coordinates of landmarks from the front view contour (in XY plane), and the z-coordinate from the side view contour(in YZ plane).

Step 4. Find out the left armpit point p_{left_armpit} and p_{right_armpit} with the following algorithm:

a) From C_f, set the target zone as $x \in [0.2h, h]$ and $y \in [0.5h, 2.5h]$.

b) Find out the point with highest y-coordinate in this zone as the left armpit point p_{left_armpit}, then form a horizontal line from p_{left_armpit} in YZ plane, intersect with C_{sAxis} at a point whose z value will be taken as the z value of p_{left_armpit} in 3D space, as shown in Fig.2.

c) Set the target zone as $x \in [-h, -0.2h]$ and $y \in [0.5h, 2.5h]$, repeat step 4b to find out the right armpit point p_{right_armpit}, then form a horizontal line from p_{right_armpit} in YZ plane, intersect with C_{sAxis} at a point whose z value will be taken as the z value of p_{right_armpit} in 3D space, as shown in Fig.2.

Step 5. Find out the left acromion point $p_{left\_acromion}$ and $p_{right\_acromion}$ with the following algorithm:

a) From C_f, set a point g as $g(x) = 0.8h$, $g(y) = 2.5h$ *.

b) Find out the point that is closest to g as the left acromion $p_{left\_acromion}$, then form a horizontal line from $p_{left\_acromion}$ in YZ plane, intersect with C_{sAxis} at a point whose z value will be taken as the z value of $p_{left\_acromion}$ in 3D space, as shown in Fig.2.

* In the following sections, we use $p(x)$, $p(y)$ and $p(z)$ to represent the x,y and z coordinates of point p

c) Set g as $g(x) = -0.8h$, $g(y) = 2.5h$. Find out the point that is closest to g as the left acromion $p_{right\_acromion}$, then form a horizontal line from $p_{right\_acromion}$ in YZ plane, intersect with C_{sAxis} at a point whose z value will be taken as the z value of $p_{right\_acromion}$ in 3D space, as shown in Fig.2.

Step 6. Separate C_f into torso, left leg, right leg, left arm and right arm with the following algorithm:

a) Take $p_{left\_acromion}$ as the start point and sort C_f in contour-clockwise direction.

b) The point's index between $p_{right\_acromion}$ and p_{right_armpit} will be categorized as right arm contour, denoted as C_{RArm}. The point's index between p_{left_armpit} and the end point will be categorized as left arm contour, denoted as C_{LArm}.

c) Except for C_{LArm} and C_{RArm}, the points with x and y coordinates smaller than that of p_{crotch} will be categorized as right leg contour, denoted as C_{RLeg}. Except for C_{LArm} and C_{RArm}, the points with x-coordinate bigger than that of p_{crotch} and y-coordinate smaller than that of p_{crotch} will be regarded as left leg contour, denoted as C_{LLeg}

d) Except for C_{LArm}, C_{RArm}, C_{LLeg} and C_{RLeg}, the points on C_f will be regarded as torso contour, denoted as C_{Torso}.

e) For each segment contour, find out the point's index with the lowest y-coordinate. The indices lower than this index will be the left part of this contour, and the indices bigger than this index will be the right part of this contour. The left and right status will be very helpful in building the correct correspondence for mean value deformation, as detailed in the following section.

Step 7. Separate C_s into torso and leg with the following algorithm:

a) The points of C_s with the same y-coordinates as that of C_{torso} should belong to side-view torso contour C_{sTorso}. The points of C_s with the same y-coordinates as that of C_{LLeg} should belong to side-view leg contour C_{sLeg}.

b) For each point p on C_{sTorso} and C_{sLeg}, find out a point p_{sAxis} with the same y value on C_{sAxis}. For $p(z) \geq p_{sAxis}(z)$, mark its status as 'front', and for $p(z) < p_{sAxis}(z)$, mark its status as 'back'.

Step 8. Set the front/back status of points on C_{Arm} with the following algorithm:

a) Take $p_{shoulder}$ as the start point and sort C_{Arm} in counter-clockwise direction.

b) Find out the point's index with the lowest y-coordinate. The points of C_{Arm} with indices lower than this index will be the front part of this contour, and the points of C_{Arm} with indices bigger than this index will be the back part of this contour.

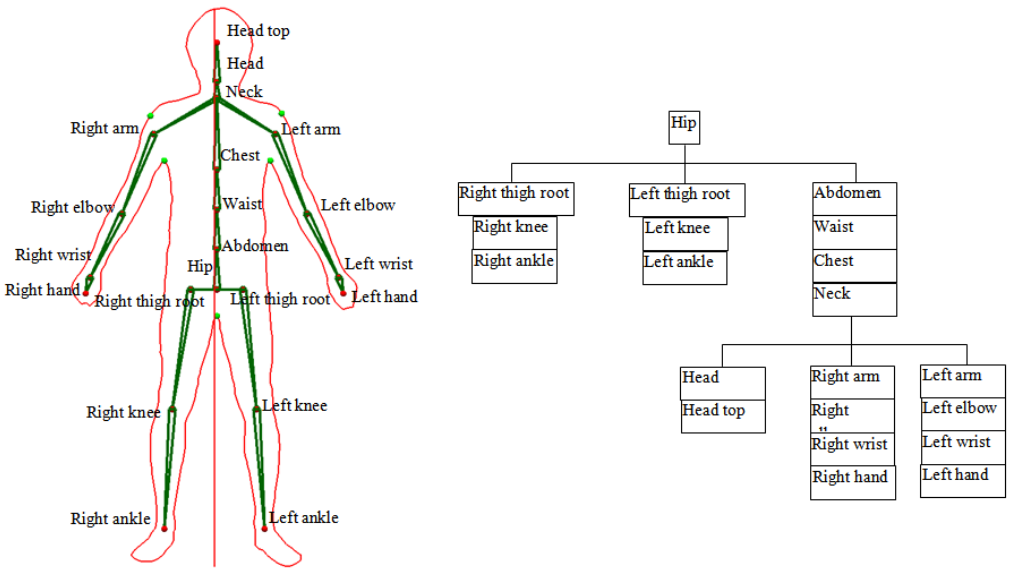

Fig. 3. The skeleton of the image contours in 3D space (from left to right): a) Illustrating the bones and joints. b) Hierarchical structure of the skeleton

Step 9. Compute the joints (as shown in Fig.3) to build the skeleton with the following algorithm:

a) Set the joint named 'Hip', whose coordinates are $x=p_{crotch}(x)$, $y = p_{crotch}(y) + 0.2h$, and $z = p_{crotch}(z)$.

b) The spine joints are first computed in XY plane at various y positions. Once the x, y coordinates have been determined, a horizontal line is formed to pass through this point to find out the intersection point with C_{sAxis}. After getting the intersection point, take its z-coordinate for the current spine joint. The x and y coordinate for each spine joint is shown in Table 1.

Table 1. Set the x and y coordinate for spine joints.

Name	x-coordinate	y-coordinate
Abdomen	$p_{crotch}(x)$	$p_{crotch}(y)+0.5h$
Waist	$p_{crotch}(x)$	$p_{crotch}(y)+h$
Chest	$p_{crotch}(x)$	$p_{crotch}(y)+1.5h$
Neck	$p_{crotch}(x)$	$p_{crotch}(y)+2.4h$
Head	$p_{crotch}(x)$	$p_{crotch}(y)+2.6h$
Head top	$p_{crotch}(x)$	$p_{crotch}(y)+3.0h$
Left thigh root	$p_{crotch}(x)+0.33h$	$p_{crotch}(y)$
Right thigh root	$p_{crotch}(x)-0.33h$	$p_{crotch}(y)$

c) Given a series of horizontal lines $y = y_c$ intersect with C_f, the mid-points can be taken as the limb joints. Once the x, y value has been determined, a horizontal line is formed to pass through this point to find out the intersection point with C_{arm}. After getting the intersection point, take its z value for the current limb joint. The y value for each horizontal line is shown in Table 2.

Table 2. The y-coordinate for each horizontal line to get corresponding limb joint.

Name	y-coordinate	Name	y-coordinate
Left knee	$p_{crotch}(y)-1.5h$	Left elbow	$p_{crotch}(y)+1.4h$
Right knee	$p_{crotch}(y)-1.5h$	Right elbow	$p_{crotch}(y)+1.4h$
Left ankle	$p_{crotch}(y)-3.0h$	Left wrist	$p_{crotch}(y)+0.6h$
Right ankle	$p_{crotch}(y)-3.0h$	Right wrist	$p_{crotch}(y)+0.6h$
Left arm	$p_{crotch}(y)+2.4h$	Left hand	$p_{crotch}(y)+0.5h$
Right arm	$p_{crotch}(y)+2.4h$	Right hand	$p_{crotch}(y)+0.5h$

2 Mean value deformation

To conduct the deformation smoothly, we employ the mean value encoding/decoding algorithm. In the procedure of encoding, the mean value coordinates for each vertex of the candidate model are computed in terms of a convex combination of its neighboring vertices, as explained in [22].

Before mean value encoding, the candidate model and the image contour are all calibrated to the same height, which is to maintain the priority of shape approximation. Once the deformation has been fulfilled, the height value represented by the original image contour or input by the online shopper will be the benchmark of scaling the new-born avatar.

Denote v as the body vertex in 3D and $v_1, v_2, \ldots, v_m$ as its neighboring vertices, the mean value encoding can be performed as:

a) Compute the projection plane $P = n_x x + n_y y + n_z z + d$, where the normal n is computed as $n = \dfrac{\sum_{i=1}^{m}(v_{i+1}-l)\times(v_i-l)}{\|\sum_{i=1}^{m}(v_{i+1}-l)\times(v_i-l)\|}$, and $l = \dfrac{1}{m}\sum_{i=1}^{m} v_i$. The average distance d from origin is computed as $d = -\dfrac{1}{m}\sum_{i=1}^{m} n\cdot v_i$.

b) Project v and all its neighbors onto the projection plane:

$$v' = v - (d + (v\cdot n))n$$

$$v_i' = v_i - (d + (v_i\cdot n))n$$

c) Compute the mean-value weight of v with respect to v_i in both tangent direction (denote as coefficient w_i) and normal direction (denote as coefficient b_i)

$$w_i' = \frac{\tan(\phi_i/2) + \tan(\phi_{i+1}/2)]}{\|v_i' - v'\|},$$

$$w_i = \frac{w_i'}{\sum_{i=1}^{m} w_i'} \qquad (2)$$

where $\phi_i = a\cos\left(\dfrac{v_i'-v'}{\|v_i'-v'\|}\cdot\dfrac{v_{i-1}'-v'}{\|v_{i-1}'-v'\|}\right)$, $\phi_{i+1} = a\cos\left(\dfrac{v_i'-v'}{\|v_i'-v'\|}\cdot\dfrac{v_{i+1}'-v'}{\|v_{i+1}'-v'\|}\right)$,

$$b_i = \frac{c_i}{\sqrt{1-c_i^2}} \tag{3}$$

where $c_i = \frac{(v-v_i)\cdot n}{\| v-v_i \|}$.

With the mean value encoding, the position of v after deformation (denoted as v_{mv})can be expressed by the local neighbors while maintaining the overall shape of the original model, say the candidate model. The decoding is computed as

$$v_{mv} = v' + \sum_{i=1}^{m} w_i[\| v' - v_i' \| b_i + (v_i - v_i')\cdot n]n \tag{4}$$

where $v' = \sum_{i=1}^{m} w_i \{v_i - [d + (v_i \cdot n)]n\}$.

For the implementation requiring shape preserving, the mean value decoding should minimize the following functional $\frac{1}{2}\sum_{v_i \in \{V\}} (v_i - v_{i\_mv})^2$, where v_i is the ith body vertex before mean value decoding, v_{i_mv} is its position after mean value decoding, and $\{V\}$ is the set of body vertices. This problem can be solved by Levenberg-Marquardt minimization [25].

A major challenge of mean value deformation is to find out the target position of the control vertices. This can be described as a problem in finding out the correspondence between body vertices and points on image contour, as shown in Fig. 4. For the convenience of explanation, we use 'vertex' to refer to the 3D points on the candidate body, and 'point' to refer to the 2D points on the image contour in the rest of this paper. The "control vertices" represents the vertices with target positions defined by the image contours, as shown in Fig.4c.

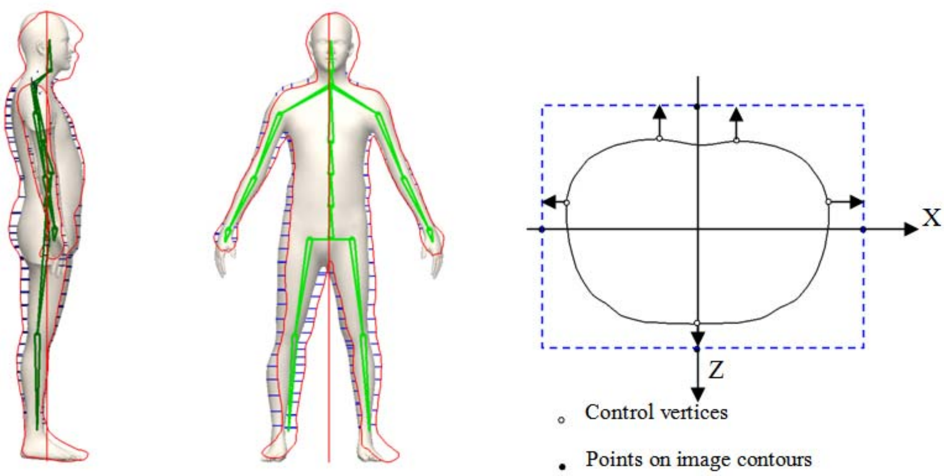

Fig.4 Finding the correspondence between the candidate model and the image contours (from left to right): a) Side view. b) Front view. c) Setting the target positions for the control vertices.

From the nature of taking the orthogonal pictures, it is easy to understand that the orthogonal contours are the representation of the outmost points on the human body, which is not always located in the sagittal/coronal plane. This indicates we cannot use the vertices along sagittal/coronal plane as the control vertices directly. In this sense, finding the correspondence is actually a problem in finding body vertex v that satisfies one of the following four equations

$$v = \arg\min_{v\in\{S\}}(x),\ v = \arg\max_{v\in\{S\}}(x),\ v = \arg\min_{v\in\{S\}}(z),\ v = \arg\max_{v\in\{S\}}(z),$$

where $\{S\}$ is the current slice contour of the candidate model. Benefit from our slice-based interpolation, the maximum and minimum values for each slice is easier to obtain. However, without loss generality, an unorganized triangle mesh can also use this slice-based approximation to find out the control vertices.

Step 1. For a given model, perform the body segmentation/measurement to get the skeleton.

Step 2. Compute the average edge length of the model, denoted as L.

Step 3. Perform volume slicing as explained in section 2C, while the interval between each slice is set to L.

Step 4. For each slice $\{S\}$, find out the point with minimum/maximum x-coordinate and z-coordinate respectively, and then find out the closest vertex to these points as the control vertices. For a control vertex that is close to $\arg\min_{p_i\in\{S\}}(x)$, mark its status as 'left', and for a control vertex that is close to $\arg\max_{p_i\in\{S\}}(x)$, mark its status as 'right'. For a control vertex that is close to $\arg\min_{p_i\in\{S\}}(z)$, mark its status as 'back', and for a control vertex that is close to $\arg\max_{p_i\in\{S\}}(z)$, mark its status as 'front'. The x-coordinate of the control vertex is determined by passing a segment from this point to get the intersection with the image contour ($C_{LArm}/C_{RArm}/C_{LLeg}/C_{RLeg}/C_{Torso}$)in XY plane, as shown in Fig. 5. The same computation is performed in YZ plane with image contour C_{sTorso} or C_{sLeg} to get the z value as the target coordinate.

As shown in Fig.5, The proposed algorithm can ensure that the control vertices always get the intersection points on the contour with same left/right or front/back status to avoid setting up wrong correspondence, which is actually imposing the positional and directional constraints respectively.

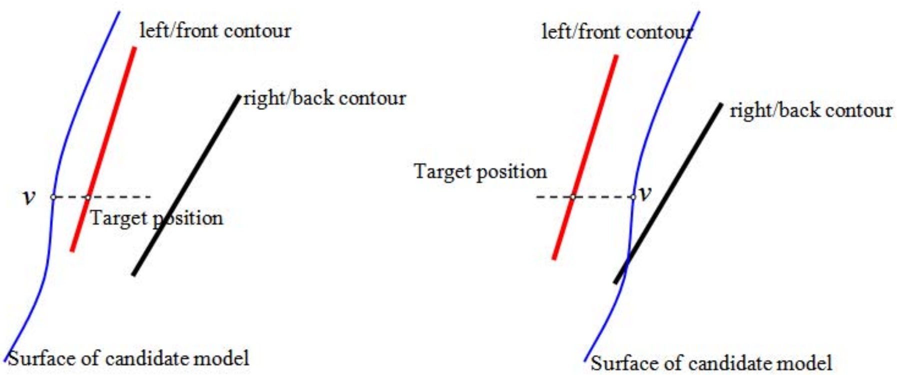

Fig.5 Relationships between control vertex v and image contour (from left to right): a) v is located outside of the image contour. b) v is located inside of the image contour.

From our experience, to set up the correspondence automatically, the pose of the candidate model should be synchronized with the pose revealed by the image contours. With the method introduced in section 2B and section 2E, this can be fulfilled conveniently. Given a set of control vertices $\{V_c\}$, the positions of the rest of the vertices on the candidate model are computed by iterating the following scheme:

a) For each vertex v, if v is not in $\{V_c\}$, update the position of v using Equation 4.

b) For each control vertex v in $\{V_c\}$, move it to the target position directly.

c) Repeat until convergence.

The convergence of the deformation is defined by observing the position change rate ε, defined as

$$\varepsilon = \frac{\sum_{v_i \in \{V\}} \| v_i - v_{i\_mv} \|}{\sum_{v_i \in \{V\}} \| v_i \|}$$

If ε is lower than a given threshold, the deformation is declared to be convergent. From our practice, setting this threshold as 1% will be a rational choice for deforming the candidate model against the image contour properly.

3 Results and discussion

In order to verify the accuracy of our proposed method, we measured the subject shown in Fig.1 at his chest, waist, abdomen, hip and thigh to extract the girth value and compared them with the values measured from the generated avatar (as shown in Fig.6) at the same places. The deviation is around ± 2 mm~± 5 mm (1mm equals 1.2 pixel in the original image). Technically, if there are 1 pixel offset in $\pm X$ and $\pm Z$ direction respectively, the girth will be at least 3.3 mm bigger or

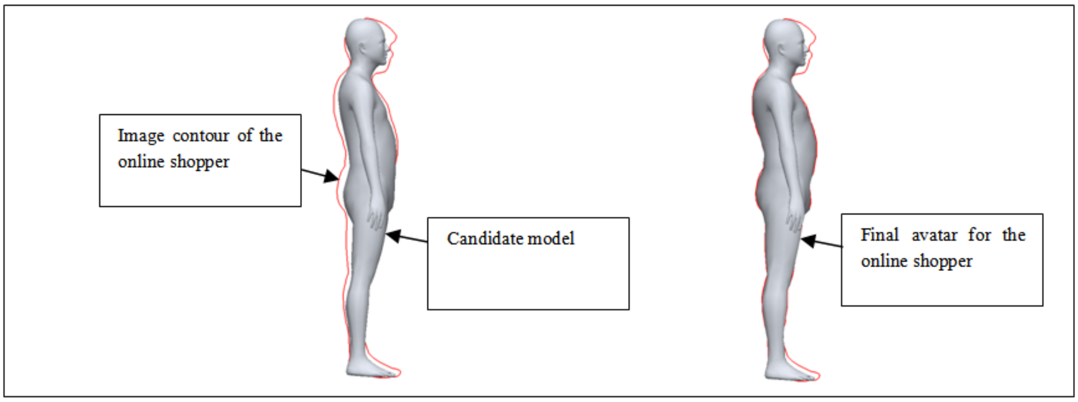

Figure 6. Shape deformation via mean value decoding against orthogonal image contours

smaller than the real one. This is probably caused by the quality of the images (whether it is orthogonal or not) or the edge detection algorithm. If the photos were poorly taken with significant distortion, the shape matching and the mean value deformation would not provide the correct answer. The height calibration/rescaling may also introduce errors. If the height value was not measured accurately, the avatar would not be scaled to the real size. These possible inaccuracies are the same dilemma as if the online shopper could not take the key measurement accurately. However, we believe that a simple set of image taken with the output as shown in Fig.6 is acceptable since the majority of the body features and sizes have been reproduced. To further verify our proposed method, we used the information obtained from internet [24] to generate a female avatar, as shown in Fig. 7, which also proves the effectiveness of our approach.

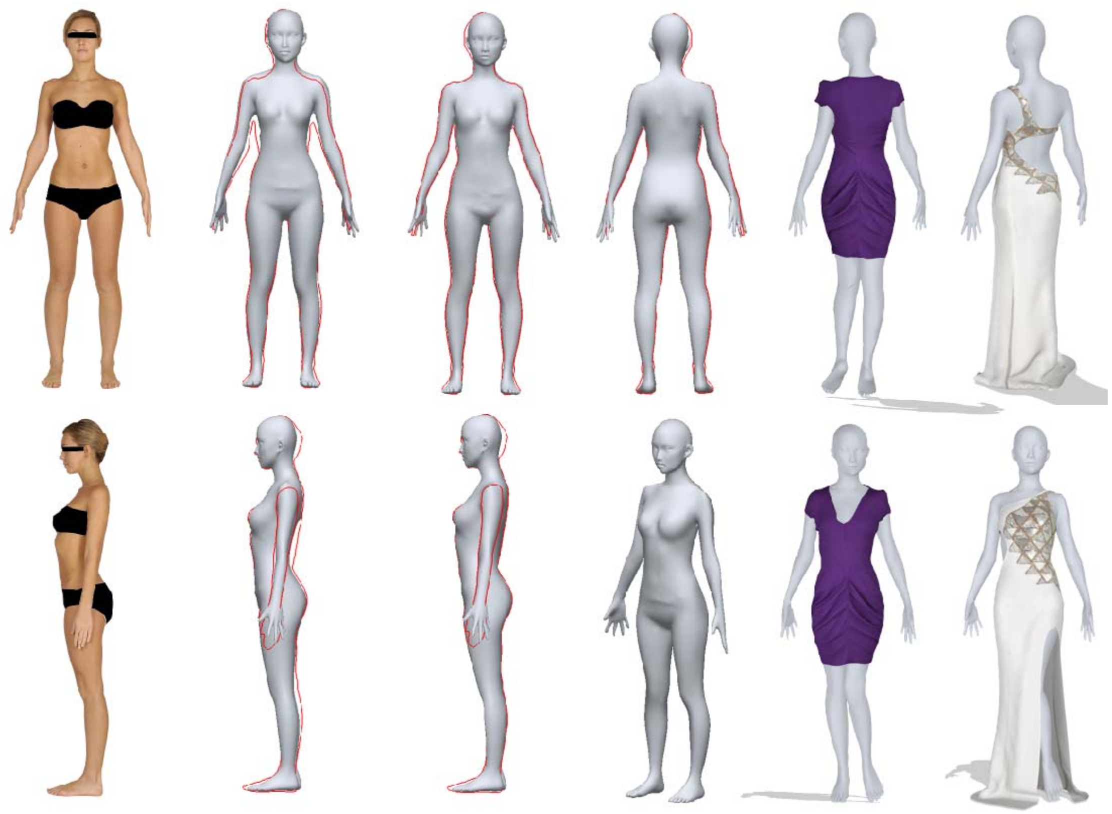

Figure 7. Generating female avatar from orthogonal images for virtual dressing. Top row, from left to right: a) Original image in front view. b) Candidate model in front view. c) Final avatar in front view. d) Final avatar in back view. e) Virtual dressing result #1 in back view. f) Virtual dressing result #2 in back view; Bottom row, from left to right: g) Original image in side view. h) Candidate model in side view. i) Final avatar in side view. j) Final avatar in 45° view. k) Virtual dressing result #1 in front view. l) Virtual dressing result #2 in front view.

The speed of generating an avatar from the given image is summarized in Table 3. The mesh density of all the template/candidate models is around 15k triangles. The test is performed on a PC with Intel(R) Core(TM) 2 Duo CPU, and 4GB physical memory. The resolution of the orthogonal images is set to 1024×1024 pixels.

Table 3. The speed of generating an avatar from the given image contours.

Shape interpolation	Candidate model selecting	Image-skeleton extraction	Mean value encoding	Correspondence setting up	Mean value decoding
33.328s	0.148s per silhouette comparison	0.039s	0.292s	0.189s	2.351s

As shown in Table 3. Most of the time is consumed by building up the shape library. The selection of the best candidate model is also time-consuming if the library has numerous template models. In our practice, the shape library is consists of 124 male subjects and 128 female subjects respectively. The overall time cost in generating the avatars shown in Fig.6 and Fig.7 is less than 1 minute respectively.

The proposed shape interpolation does not require a pre-marking procedure compared to the work of Allen et al [9] and Anguelov et al [13]. The pose of the range data can be different from each other. The pose synchronization procedure will minimize the difference for a smooth morphing. This indicates that we can sample from various scanned range data set and still provide a satisfied result automatically.

Reconstructing the face, hair and fingers/toes is beyond the scope of this paper. These regions are regarded as rigid component and are transformed rigidly to maintain its connectivity with the human body, as shown in Fig.6 and Fig.7. The orthogonal images shot for the whole-body-deformation may not posses the required resolution for a clear representation to morph these parts appropriately. Furthermore, it is a very challengeable task to recover the head and fingers/toes from two still images automatically. The same scenario happens to the full body texturing where generating the texture atlas automatically requires a robust approach in surface parameterization, image shading and stitching. Our future work will be focused on these topics.

4 Conclusion

In this paper, an image based method is proposed to generate an avatar for the online shopper when 3D body scanning is not feasible. The online shopper is required to provide two orthogonal images as the target silhouettes. The candidate model is then morphed against the image contours via mean value deformation to generate the final avatar for the online shopper.

The major contribution of this work is to provide a marker-free approach in generating the avatar from user-input orthogonal images without any further requirement. The results validate that the presented method is a remarkable treatment in producing high fidelity body shape for online shopper.

References

[1] LEWIS JP, CORDNER M and FONG N. Pose space deformation: A unified approach to shape interpolation and skeleton-driven deformation, SIGGRAPH'00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques[M], New Orleans, Louisiana, USA, 23-28 July, 165～172. ACM Press/Addison-Wesley Publishing Co., New York, 2000.

[2] SINGH K, and KOKKEVIS E. Skinning characters using Surface-Oriented Free-Form deformations, Proceedings of the Graphics Interface 2000[M], Montréal, Québec, Canada, 15-17 May, 35～42. Canadian Human-Computer Communications Society 2000, Québec.

[3] SLOAN PPJ, ROSE CF, and COHEN MF. Shape by example. Symposium on Interactive 3D Graphics[M], NC, 19-21 March, 135～143, ACM, New York, 2001.

[4] CAPELL S, GREEN S, CURLESS B, DUCHAMP T and POPOVIĆ Z. Interactive Skeleton-Driven Dynamic Deformations, ACM Transactions on Graphics (TOG)[J], 2002, 21(3):586～593.

[5] LEE A, DOBKIN D, SWELDENS W, and SCHRIDER P. Multiresolution Mesh Morphing. SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques[M], Los Angeles, CA, 8-13 August, 343～350. ACM Press/Addison-Wesley Publishing Co., New York, 1999.

[6] LIN JL, CHUANG JH, LIN CC and CHEN CC. Consistent parametrization by quinary subdivision for remeshing and mesh metamorphosis. Proceedings of the 1st international conference on Computer graphics and interactive techniques in Australasia and South East Asia[M], Melbourne, 11-14 February, 151～158. ACM, New York, 2003.

[7] PRAUN E, SWELDENS W and SCHRÖDER P. Consistent Mesh Parameterizations. SIGGRAPH '01, Proceedings of the 28th annual conference on Computer graphics and interactive techniques[M], Los Angeles, California , 12-17 August, 179～184. ACM, NY, 2001.

[8] ALLEN B, CURLESS B and POPOVIC Z. Articulated body deformation from range scan data. SIGGRAPH '02, Proceedings of the 29th annual conference on Computer graphics and interactive techniques[M], San Antonio, Texas, 21-26 July, 612～619. ACM, NY, 2002.

[9] ALLEN B, CURLESS B and POPOVIC Z. The space of human body shapes: reconstruction and parameterization from range scans. ACM Trans. Graphics (TOG)[J], 2003, 22(3): 587～594.

[10] HILTON A, STARCK J and COLLINS G. From 3D shape capture to animated models. Proc. First International Symposion on 3D Data Processing, Visualization, and Transmission (3DPVT 2002)[M], Padova, Italy, 19-21 June, 246～255, IEEE Computer Society, NY, 2002.

[11] PAULY M, MITRA NJ, GIESEN J, GROSS M and GUIBAS L. Example-based 3D scan completion. Symposium on Geometry Processing[M], Vienna, Austria, 4-6 July, pp. 23～32, ACM, NY, 2005.

[12] KRAEVOY V, SHEFFER A. Cross-Parameterization and Compatible Remeshing of 3D Models. ACM Trans. Graphics (TOG)[J], 2004, 23(3): 861 ～ 869.

[13] ANGUELOV D, SRINIVASAN P, KOLLER D, THRUN S, RODGERS J, and DAVIS J. SCAPE: Shape Completion and Animation of People. ACM Trans. Graphics (TOG)[J], 2005, 24(3): 408～416.

[14] MOHR, A, and GLEICHER, M. Building Efficient, Accurate Character Skins from Examples. ACM Trans. Graphics (TOG)[J], 2003, 22(3): 562～568.
[15] ZIMMERMANN J, NEALEN A, and ALEXA M. Sketching contours. Computers & Graphics[J], 2008, 32(5): 486～499.
[16] SORKINE O, COHEN-OR D, LIPMAN Y, ALEXA M, RÖSSL C, and SEIDEL H. Laplacian surface editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing[M] (Nice, France, July 08 - 10, 2004). SGP '04, 71:175～184, 2004.
[17] PRASAD M and FITZGIBBON A. Single View Reconstruction of Curved Surfaces. In Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 [M](June 17 - 22, 2006). CVPR. IEEE Computer Society, Washington, 1345～1354, 2006.
[18] YANG C, SHARON D, and VAN DE PANNE M.. Sketch-based modeling of parameterized objects. In ACM SIGGRAPH 2005 Sketches[M] (Los Angeles, California, July 31 - August 04, 2005). J. Buhler, Ed. SIGGRAPH '05. ACM, New York, NY, No.89 , 2005.
[19] KHO Y and GARLAND M. Sketching mesh deformations. In Proceedings of the 2005 Symposium on interactive 3D Graphics and Games[M] (Washington, District of Columbia, April 03 - 06, 2005). I3D '05. ACM, New York, NY,147～154, 2005.
[20] KRAEVOY V, SHEFFER A, and VAN DE PANNE M. Modeling from contour drawings. In Proceedings of the 6th Eurographics Symposium on Sketch-Based interfaces and Modeling[M] (New Orleans, Louisiana, August 01 - 02, 2009). D. Fellner and S. Spencer, Eds. SBIM '09. ACM, New York, NY, pp.37～44, 2009
[21] ZHONG, YQ, and XU, BG. Automatic Segmenting and Measurement on Scanned Human Body. International Journal of Clothing Science and Technology[J], 2006, 18(1):19～30.
[22] KRAEVOY V, and SHEFFER A. Mean-value geometry encoding. International Journal of Shape Modeling[J], 2007, 12(1): 29～36.
[23] RATNER P. 3-D Human Modeling and Animation[M], 2nd ed., Wiley, New York, NY, 55～57, 2003,
[24] http://www.cgrealm.org/gallery/maps/photo/200807/19-891.html
[25] WILLIAM H. PRESS, BRIAN P. FLANNERY, SAUL A. TEUKOLSKY, and WILLIAM T. VETTERLING. Numerical Recipes: The Art of Scientific Computing[M]. Cambridge University Press, Cambridge (UK) and New York, 2nd edition, 1992.

基于正交图像轮廓的三维人体自动重建

钟跃崎[1,2]
（1. 东华大学纺织面料技术教育部重点实验室，上海 201620；
2. 东华大学纺织学院，上海 201620）

摘要：随着服装在线销售的增长，人体形态成为了虚拟空间中衡量服装着装效果的一个重要基石。本文提出了一个根据在线购物者所提交的两张正交轮廓照片，对已有的模板人体进行变形，从而自动重建消费者三维化身的算法。该模板人体来自于真实的三维人体扫描数据，采用均值编码变形法对该人体进行变形，通过施加位置和方向性约束，自动建立了模板人体表面的控制点与正交图像轮廓上对应点间的关联。根据该方法，可以有效地生成代表在线购物者特征的三维人体。

关键词：计算机图象形学；三维人体；二维照片轮廓；均值编码变形

中图分类号：TP391.41