Modeling and Deformation Method of Human Body Model Based on Range Data

Ryouji Matsuda
Faculty of Engineering
Fukuyama University
Higashimura-cho, Fukuyama, Hiroshima,
729-0292 Japan
matsuda@toubu-kg.pref.hiroshima.jp

Tomuyuki Nishita
Faculty of Science
Tokyo University
7-3-1Hongo,Bunkyo-ku,Tokyo,
113-0033 Japan
nis@is.s.u-tokyo.ac.jp

Abstract

A cloth simulation system must generate a human body model based on measured data obtained from range data. We propose modeling and deformation methods based on such data. In our system, the human body is modeled by layered metaballs which correspond to the horizontal cross section of the body. For each cross section, metaballs are generated by measured sample points on the boundary of the cross section. In order to fit the metaball surface with the sampling points, we employed the Steepest Descent method. For body deformation, the sampling points on the cross section are smoothly moved using Bezier curves. To show the effectiveness of the proposed method, we demonstrate fitting and deformation, the two human body models to be used for the cloth simulation.

1. Introduction

One application of computer graphics is cloth simulation or CAD system for cloth design. Such a cloth simulation system must generate human body models based on measured data obtained from range data for a given person putting on clothes. Metaballs are useful for modeling a human body because of the smooth and flexible surfaces. In our system, the human body model is created by layered metaballs which correspond to the horizontal cross section of the body. This paper proposes two methods: a method for fitting the metaball surface to the measured surface, and a body deformation method.

In our approach, we prepare a basic human body modeled by metaballs. This model can be optimized to fit the measured human body. For each cross section of the human body, metaballs are generated using sampling points on the boundary of the cross section (i.e., contour curve). In order to fit the metaball surface to the sampling points, we employed the Steepest Descent method. We optimize the parameters (center position, radius and density) of the metaballs by minimizing the density error evaluated at each sampling point on the cross section.

In some cases, we need interactive deformation of the human body model. We propose a deformation method using Bezier curves for smooth deformation. That is, the sampling points on each cross section are deformed by the Bezier curves controlled by a few mouse operations. For this operation, we propose an effective algorithm for calculating the closest point on a Bezier curve using the Bezier Clipping method [1] which is an iteration method using only linear equations. Previous methods for finding the closest point on a curve have depended on two calculation methods: the sampling method (finding the closest point within subdivided curves) and the numerical method [2] using the dot product of the derivative of the curve and the vector from the point to the curve. These methods insufficient in light of computational cost, so we propose a more effective approach.

Our goal is to create a human body model for cloth simulation. Thus, using the proposed fitting method, we have developed two human body models, the standard human model from the measured data of the average Japanese and a deformation of that. To show the effectiveness of the proposed method, cloth simulation is applied to these models.

2. Fitting metaballs to the measured body using the Steepest Descent method

2.1. Object representation by metaballs

In the metaball technique, a surface of metaballs is defined as the iso-surface (equi-potential surface) of a field function. We use the field function proposed by Nishimura [3].

The field value at any point (x, y, z) is defined by distances d_i from the specified center position of each metaball i in three-dimensional space. The task of the user is to specify the center position (x_i, y_i, z_i) of each metaball i, its density w_i at the center, effective radius r_i, and field function $f(d_i)$. For n metaball, the density W at point (x, y, z) is defined by

$$W(x,y,z) = \sum_{i=1}^{n} w_i f(d_i). \qquad (1)$$

A surface of metaballs is defined as an iso-surface when the density equals threshold value T. If there is only one metaball, the shape is a sphere. Threshold radius r_t is defined as the radius of the metaball when the density equals the threshold value.

2.2 Using the Steepest Descent method

A cloth simulation system must generate human body models based on measured data. Muraki [4] proposed a "Blobby

Model" for automatically generating a shape description from range data. He started with a single metaball and introduced more metaballs by splitting each metaball into two further metaballs so as reduce the energy value. However, the 3D object is slowly recovered as the iso-surface produced by a large number of metaballs. Bitter [5] proposed a method that combined medial axises and implicit surfaces in reconstructing a 3D solid. However, this method requires many implicit primitives. If the model is constructed of too many implicit primitives, the computational cost for cloth simulation becomes prohibitively expensive. To generate the human body model with a small number of metaballs and minimal error from the mesured data, we propose generating the model by using layered metaballs which correspond to the horizontal cross section of the body.

Cross sections of the body are extracted from measured data in equal space. Metaballs are arranged roughly inside the contour of each cross section by hand (see Fig.1). The relationship between the sampling points on a cross section of body and themetaballs is shown in Fig.2. The metaball parameters (x_i, y_i, z_i, r_i, w_i) are optimized by minimizing the density error evaluated at each sampling point on the contour of the cross section of the body.

Field densities for each metaball (based on distance between the sampling point and each metaball) are summed. When the density on the sampling point arrives at the threshold density T, the metaball surface fits the sampling point perfectly. The square-sum of error between the density on each sampling point j and threshold value T is defined by

$$E = \frac{1}{2}\sum_{j=1}^{m}\left(\sum_{i=1}^{n} w_i f\left(d_{ij}\right) - T\right)^2, \qquad (2)$$

where d_{ij} is the distance from the sampling point j to the center of metaball i, *and* n is the number of metaballs.

We optimize the metaball parameters (x_i, y_i, z_i, r_i, w_i) by minimizing E evaluated at all sampling points on the cross section. Non-linear optimization is necessary to solve this problem. Though any non-linear optimization method can optimize the parameters, to optimize the parameters by the Newton method or Quasi-Newton methods (to generate a human body model), an inverse matrix of enormous unknown numbers (i.e., 5n) must be calculated at every iterating step. For this optimization, we employ the Steepest Descent method because of its simple algorithm. Parameter $X^{(0)}$ for the initial shape of the metaballs is defined by

$$X^{(0)} = (x_1, y_1, z_1, r_1, w_1, ..., x_n, y_n, z_n, r_n, w_n)^t, \qquad (3)$$

where t is the transposition symbol.

Gradient ∇E of E is calculated in the Steepest Descent method. ∇E is defined by

$$\nabla E = \left(\frac{\partial E}{\partial x_1}, \frac{\partial E}{\partial y_1}, \frac{\partial E}{\partial z_1}, \frac{\partial E}{\partial r_1}, \frac{\partial E}{\partial w_1}, ..., \frac{\partial E}{\partial x_n}, \frac{\partial E}{\partial y_n}, \frac{\partial E}{\partial z_n}, \frac{\partial E}{\partial r_n}, \frac{\partial E}{\partial w_n}\right)^t. \qquad (4)$$

The parameter $X^{(0)}$ and $\nabla E(X^{(0)})$ are substituted into the next equation, and a new parameter $X^{(1)}$ for the metaballs and $\nabla E(X^{(1)})$ are calculated. An operation to calculate the next new metaball parameter is repeated until E becomes small enough, at which point the optimum value of the parameter is obtained.

$$X^{(k+1)} = X^{(k)} - \alpha \nabla E\left(X^{(k)}\right), \qquad (5)$$

where α is step size, k is the number of iterative steps, $X^{(k)}$ is the parameters of the metaballs, and $X^{(k+1)}$ is the new parameters of the metaballs.

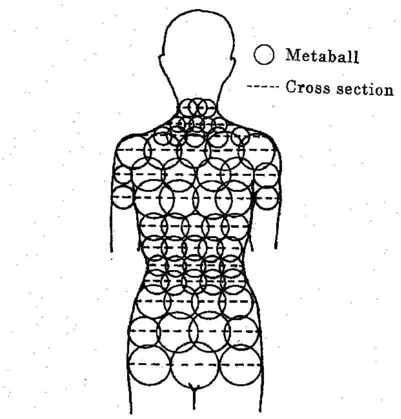


Figure 1. Combination of metaballs on cross sections.

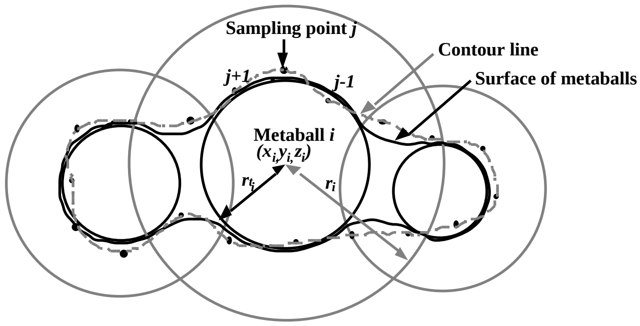
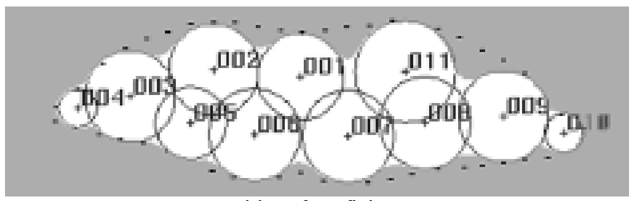
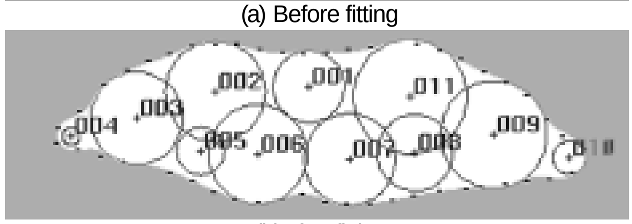


Figure 2. Relationship between sampling points on a cross section of body and metaball surface.



(a) Before fitting



(b)After fitting

Figure 3. Sampling points on cross section including acromion and metaballs.

As an example, Fig.3(a) shows sampling points on a cross section including acromion and metaballs before optimizing. Eleven metaballs are arranged roughly in order to fill up the inside of the sampling points on the contour of the cross section. The black points are sampling points, the circles are the surfaces of each metaball on the cross section, and the numbers in the circles are identification numbers for the metaballs. The metaball surface on the cross section is represented by the white area. The optimized result is shown in Fig.3 (b). As shown in this figure, the metaball surface fits the sampling points.

The metaballs on every cross section are combined (i.e. layered). After combining, the metaball surface changes slightly due to interaction between the layered metaballs. Then, the parameters of all layered metaballs are optimized. We can save computation time by using this two step optimization, one for each layer and one for the combined layers.

3. Deformation of the human body model using Bezier curves

In our system, the human body model is layered. Free-form deformation of the body can be done by deformating each layer. That is, the deformation can be achieved by the combination of 2D operations. In our approach, the shape of the human body is defined by sampling points on the cross sections. That means that the deformation can be done by moving the sampling points in a 2D plane. We propose a method of free-form deformation using Bezier curves controlled by a few mouse operations. Sampling points can be moved along every kind of curve, including B-spline curve, because any curve can be converted into Bezier curve. The main idea of the deformation is to move the sampling points through field morphing: the field is defined by Bezier curves, and these curves are deformed by detection of the closest point on the curve to a point on a screen.

3.1. Previous work for calculation method of the closest point on a curve

We have two methods to find the closest point on curve C defined by parameter u to a specified point Q.

The subdivision method into sub-spans is as follows; the method evaluates curve points at equally spaced parameter values on a curve, and compute the distance of each point from Q, and choose parameters to be the value yielding the point closest to Q. In this method, we need 10^n of point evaluations and distance calculations when we need an accuracy of 10^n (e.g., n=3) on parametric space (see Fig.4(a)).

While the numerical method is as follows; the distance from Q to a point P on curve $C(u)$ (see Fig.4(b)) is minimum when the dot product of tangent vector $\boldsymbol{C'}(u)$ and the vector $\boldsymbol{QP}$ is zero (see [2]). That is, when the vector from Q to a point P on the curve is perpendicular to the tangent vector $\boldsymbol{C'}$, the distance PQ is minimum (see Fig.4(b)). That is, we can obtain the following equation.

$$\boldsymbol{C}'(u)\ \ (C(u)-Q)=0. \qquad (6)$$

The above equation corresponds to

$$(x(u)-x_q)\frac{dx(u)}{du}+(y(u)-y_q)\frac{dy(u)}{du}=0.$$

The distance from Q to $C(u)$ is minimum when equation (6) is satisfied. In the case of degree n curve, we should solve degree ($2n$-1) polynomial. In general this is solved by Newton iteration. For Newton iteration, however, we need the initial guess for the iteration, and it is not robust for multiple roots (minimal points).

Given a point Q assumed to lie on the curve $C(u)$ of degree n, point inversion is the problem if finding the corresponding parameter, such that $C(u) = Q$. It is known that point inversion can be solved in closed form if $n \le 4$. The problem is exacerbated when the point Q is not precisely on the curve, so we need to solve it by using the projection for curves (or minimizing the distance between Q and C) .

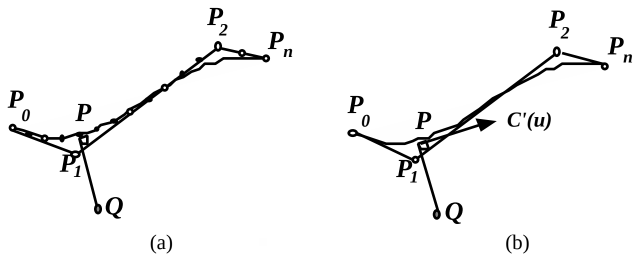


Figure 4. Finding the closest point on a curve.

3.2. Proposed method for calculation of closest point

All parametric curves can be converted to Bezier form. So we discuss here planer Bezier curves. We define a function g(u) which is the left side of equation (6), and g is given by

$$g(u)=\boldsymbol{C}'(u)\ \ (C(u)-Q), \qquad (7)$$

where

$$C(u)-Q=\sum_{i=0}^{n}(P_i-Q)B_i^n(u), \qquad (8)$$

$$\boldsymbol{C}'(u)=\sum_{i=0}^{n-1}(P_{i+1}-P_i)B_i^{n-1}(u),$$

and $B_i^n(u)=\binom{n}{i}u^i(1-u)^{n-i}$ denote the Bernstein basis polynomial.

The function $g(u)$ in equation (7) is a degree ($2n$-1) polynomial in Bernstein form. We can obtain the minimum distance when $g(u)$= 0 is satisfied. In the case of degree 3 curve, we have to solve degree 5 polynomial for which there is no closed form to solve and which has computational cost and robustness problems (These can not be solved analytically in the case of such a high degree curve). For such higher degree of curves, the problems are more serious. The method proposed here overcomes them by using the Bezier Clipping Method which was developed for ray tracing of Bezier patches [1]. The root of the function is effectively and precisely solved by using Bezier Clipping which uses the convex hull property of Bezier curves and is an iterative (and robust) method using liner equations for higher degree functions.

The method converges to roots by clipping away the intervals which have no solutions; these intervals are extracted by using the geometric characteristics of the convex hull property of Bezier curves. In this problem we could get enough of a solution through several iterations (3 to 8). This means the proposed method is 2 digits faster than the sampling method (or subdivision method) mentioned before.

Intervals containing a root or roots are extracted by using function g which is converted from C and $\boldsymbol{C'}$, and curve C is clipped by using the interval. By repeating this process, intervals containing solutions become narrow, and then the solution can be obtained.

3.3. Outline procedure of the proposed method

Lets consider an arbitrary point $Q(x_q,y_q)$ and Bezier curve $C(u)$ whose control points $P_i(x_i,y_i)$ (i=0,..,n); the derivative of curve C is called hodograph $\boldsymbol{C'}$. Example of the third curve and arbitrary point Q is shown with Fig.5 (a). Function $g(u)$ and convex closure of the control point are shown with Fig.5 (b).

The algorithm finding the closest point P on a planer Bezier curve C within a specified distance R_{max} from Q is as follows(see Fig.6):

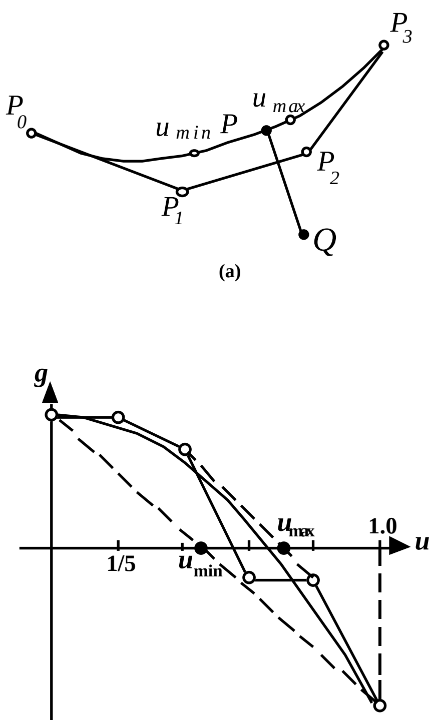


Figure 5. Extracting the interval satisfying $g(u)$=0 (in the case of cubic Bezier curve).

1) After transforming control points of C so as to be origin Q (see equation 8), calculate distances, d_0, dn, from Q to two endpoints, P_0, P_n, and set the minimum distance of d_0, d_n and R_{max} as d_{min}, that is, d_{min}=min(d_0, d_n, R_{min})

2) Clip curve C by the band which bounds of the circle with radius d_{min} and which is perpendicular to P_0P_n (see Fig.6)

3) Obtain $\boldsymbol{C'}$ from curve C (see equation 8), and calculate function g from C and $\boldsymbol{C'}$ (see equation 7)

4) If the signs of all control points of g are positive(or negative) (i.,e., there is no different sign), then this section(segment) of the curve has no solution, and so proceed to the next section, and return to step 3.

5) Extract the parameter interval which satisfies g= 0 (see [u_{min}, u_{max}] in Fig.5(b)).

6) If a Bezier clip fails to reduce the parameter interval width (u_{max}-u_{min}), split the curve in half, then return to step 3 for one half.

7) If u_{max} - u_{min} > ε (user given tolerance), clip the curve with this interval, then return to step 3.

8) Calculate distance d by assuming as u=(u_{min}+u_{max})/2, redefine the minimum distance d_{min} = min(d, d_{min}), store x and y at point $C(u)$, if there is any remaining section of the curve, return to step 3.

Note that the function g is not solved directly; g is just used for extracting the interval which has solutions (roots). Bezier Clipping is completed by subdividing C into three segments using the *de Casteljau* algorithm. The degree of g is higher than that of C, so the clipping process is performed to curve C not for g.

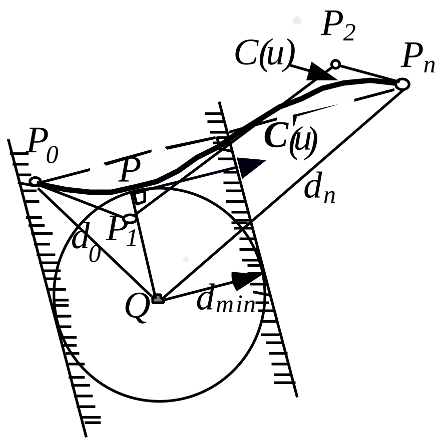


Figure 6. Clipping the Bezier curve by using the band with width d_{min} .

3.4. Interactive shape modification of curves

The designer usually modifies a parametric curve by moving one of its control points. It is difficult to predict the final shape of the curve if we move the control point. So we propose a useful technique which can deform the curve directly by moving a specified point on the curve using the mouse on a screen.

We can find parameter value u at the closest point P on a curve by clicking point Q near the curve using a mouse button. Lets consider moving point P of the curve.

After selecting the closest control point P_k to parameter value u, the displacement of control point ΔP_k can be obtained by using the displacement of mouse cursor ΔP at P (see Fig. 7).

$$\Delta P_k = \Delta P / B_k^n(u)\,. \tag{9}$$

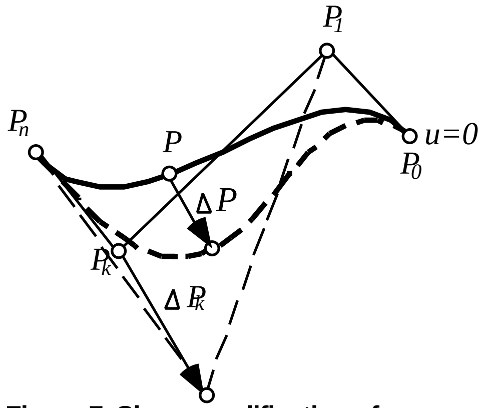


Figure 7. Shape modification of a curve.

3.5. Smooth deformation using Bezier curves

In our approach, the shape of the human body is defined by sampling points on cross sections. This means that the deformation can be done by moving the sampling points in a 2D plane. For this operation we have to solve two problems: limiting the number of operations and smoothing the connectivity between sampling points. For smooth displacement of sampling points, the idea of image warping can be applied.

For image warping, there are several methods such as feature-based method [6], mesh warping [7,8], and FFD(Free Form Deformation) [9]. The method proposed can be categorized as feature-based method.

Beier [6] has introduced a technique for morphing based upon fields of influence surrounding two-dimensional control primitives. He called this approach field morphing. He used line segments as control primitives. In our method, however, the field morphing is performed by using Bezier curves. He used two parameters, the distance from the line v, and the position along the line u ($0<u<1.$). We extend this idea to Bezier curves. If the degree of the curve is one, the both methods are equivalent. Our method also defines two parameters, u and v, as mentioned before.

Compared with the method using line segments, the method using Bezier curves has the following advantages:

(1) Deformation is performed by small number of curves.

(2) Smooth deformation (little variation of distortion) can be realized. (For smooth transition we need the number of connected lines along the curve.)

(3) Complex deformation can be done by moving only one point on the curve.

Compared with mesh warping, the method proposed here has the following advantages:

(1) Though a lot of mesh points have to be moved in mesh warping, moving only one or a few points on a curve create sufficient visual effects.

(2) For Bezier mesh (used in FFD [9]) and B-spline mesh [7], the displacement of control points do not correspond to the displacement of the image.

It is the displacement of the points around the curve corresponds to the displacement of image. Therefore, this method is very effective for interactive systems.

By using the distance calculation to curves, we can realize feature-based deformation. We can get two parameter values u and v with respect to the relationship between a point Q and Bezier curve C, where u is a parameter value at the closest point P on curve $C(u)$ from Q, v is the distance from Q to P. By using the techniques described in the previous sections, we have a useful tool. That is, these parameters can be obtained by the method in section 3.2, and we can easily deform the curve interactively by using the technique in section 3.4. As for the application of these techniques, we propose methods for deformation of object representation by metaballs.

First, we discuss the calculation method of the coordinates system related to a single curve. Lets denote parameters (u_k, v_k) for point P_k with respect to curve C_k, the coordinates of point P is defined by (see Fig.8-a)

$$P_k = C_k(u_k) + v_k N(u_k), \qquad (10)$$

where N is a unit normal vector at point $C_k(u_k)$.

After deforming curve C_k to C_k^d, point P moves to point P_k^d having the same parameters (u_k, v_k), the new point is defined by (Fig.8-b)

$$P_k^a = C_k^a(u_k) + v_k N^a(u_k). \qquad (11)$$

For m Bezier curves, $C_k(k=1,\ldots,m)$, new point P^d is given by

$$P^d = \sum_{k=0}^{m} w_k P_k^d, \qquad (12)$$

where w_k are weights which are of similar definition to those used in [6]; the weights w_k assigned to curve k should be strongest when the point is exactly on the curve, and weaker the further from it the point is.

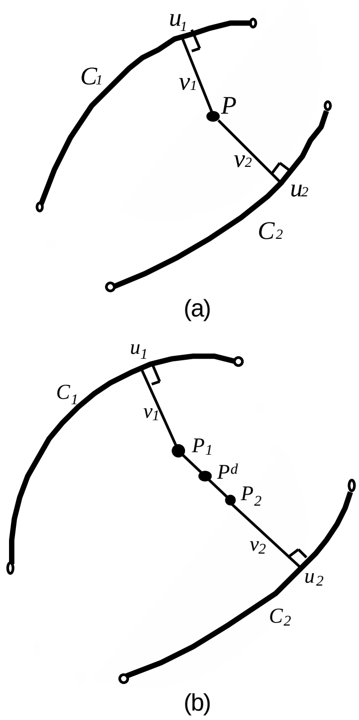


Figure 8. Transformation of coordinates by using Bezier curve.

3.6. Deformation of the human body model

So far, we have described 2D deformation. The human body model is created by layered metaballs which correspond to the horizontal cross section of the body. On each cross section, metaballs are generated from sampling points on the boundary curve of the cross section. These sampling points are deformed. The procedure for deformating the human body model is as follows.

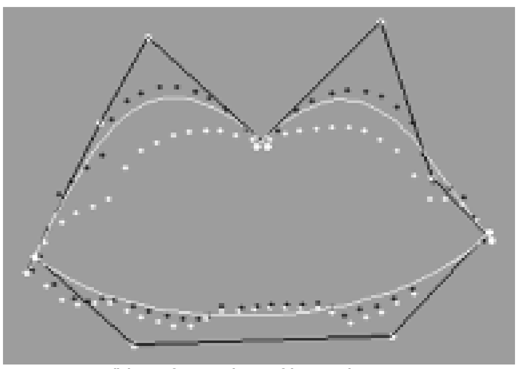
(1) Select the cross sections of the human body to be deformed after displaying the sampling points on a screen (see Fig.9(a)).

(2) Overlay Bezier curves along with the sampling points of the cross section, and deform the sampling points (red points in the figure(see color page)) by deforming the Bezier curves (see Fig.9(b)). The curves are deformed by moving the mouse.

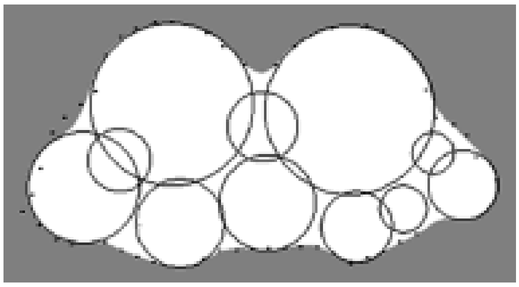
(3) Optimize the parameters of metaballs to fit the deformed sampling points (see Fig.9(c)). In this optimization, the Steepest Descent method is used (see Section 2).

Fig.9(a) shows the screens of JAVA applets which display finding the cross sections of the human body. In Fig.(b), three Bezier curves are overlaid on the sampling points on the boundary of the cross section. Two are selected by the mouse cursor and deformed. The deformation can be obtained easily with a few mouse operations. As shown in the figure, the points along(or close to) the curves are moved to the points on the deformed curves. That is, the displacement of points around the curves is exactly same as that of the mouse movement. The deformation

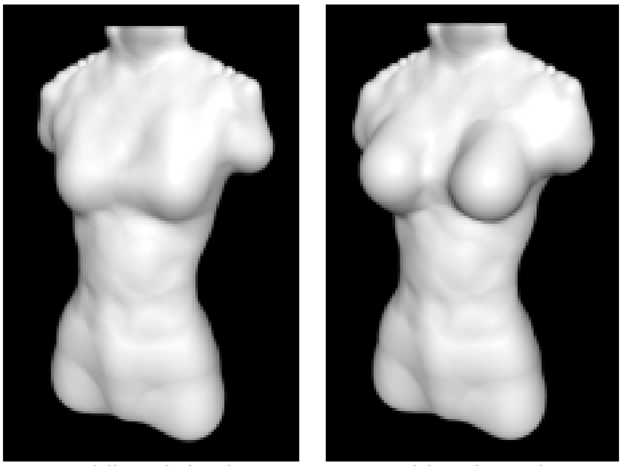
(a) Sampling points on boundary curves of horizontal cross sections



(b) Deformation of boundary



(c) Optimization of metaball



(d) Original (e) Deformed

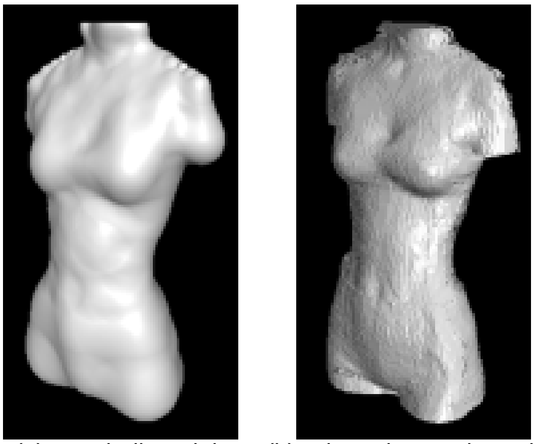
Figure 9. A human body modeled by metaballs (some sample points on the cross sections of the metaball surface are deformed by the proposed method).

is very smooth because the points move along the Bezier curve. By moving the curve, sampling points(red points in the figure(see color page)) are moved smoothly. If we want to hold some of the points, we can overlay the curves on them as constraint curves. In order to fit the surface of the metaballs to the deformed sampling points, the position and radius of each ball is optimized using the Steepest Descent method(see Fig.(c)). Fig.9 (e) shows an example of deformation (the breast is deformed). Fig.(d) is the original body.

4. Examples

Fig.10(a) shows the human body model based on the measured data of a standard Japanese female whose height is 156.8 cm and bust girth is 81.5 cm. For comparison with (a), Fig.10 shows the body model in triangular patches. Table 1 shows distance-error between the surface of metaballs and the sampling points on the major cross sections. As shown in the table, we can get the human body model with minimal error. The cpu times for first and second steps are 6 minutes and 118 minutes on a Silicon Graphics Indigo R4000, respectively. The cpu time for the method not divided into two steps is 201 minutes. Thus, the proposed method saves computation time.

To show the effectiveness of the proposed fitting and deformation methods, the two human body models are applied for cloth simulation [10,11,12]. The cloth simulation is applied to the standard human body model, then to the deformed model from the standard. Arms are added to the models in Fig.10. We can move their arms. Fig.11 shows paper patterns for a blouse simulation. Fig.12(a) shows a standard body model raising its arms. Fig.12(b) shows a body model with a breast deformed from the standard. Fig.12(c) and (d) show cloth simulations of the blouse for the standard body and the one with the deformed breast , respectively. In this case, the same blouse is used for both bodies. As shown in these images, we can get realistic shapes of clothes to fit various human bodies.



(a) Metaball model (b) Triangular patch model

Figure 10. Standard human body model.

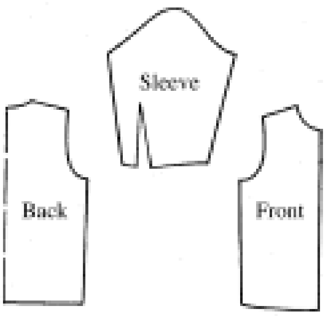
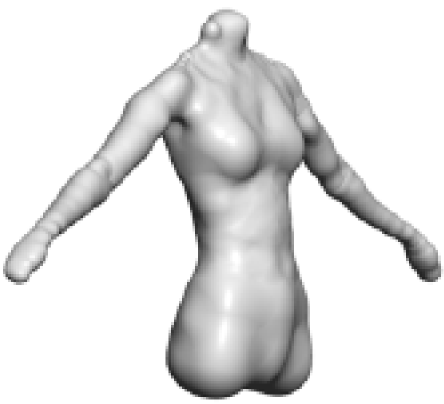
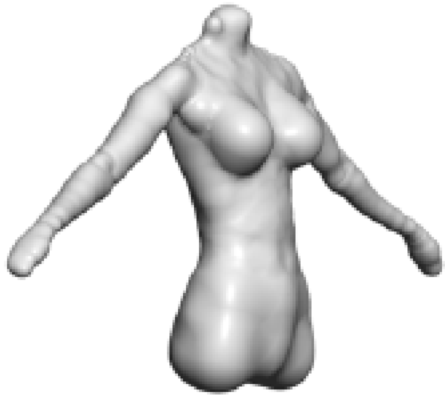


Figure 11. Paper patterns for a blouse.

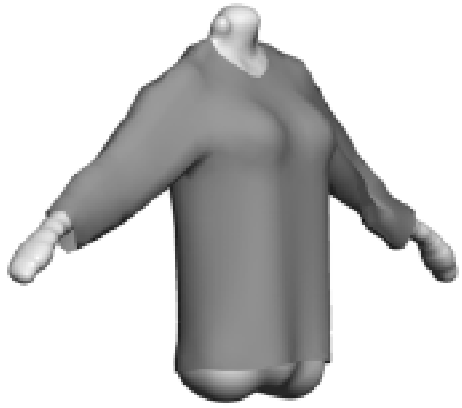


(a) Standard human body model



(b) Deformed model

(c) Simulated shape of blouse on standard human body model



(d) Simulated shape of blouse on deformed model

Figure 12. Simulated shape of the blouse on the human body model.

Table 1. Distance-error between the metaball surface and the sampling points on the major cross sections.

	Distance-error after fitting		
	Mean (mm)	S.D.	Max (mm)
Cervicale	1.57	1.13	5.24
Acromion	1.28	0.94	4.5
Nipple	0.98	0.88	4.66
Waist	0.56	0.41	1.7
Iliospinale	1.54	0.88	3.65
Hip	0.79	0.61	2.49

5. Conclusion

A cloth simulation system must generate human body models based on measured data obtained from range data. This paper has proposed modeling and deformation methods of human body. This paper has also proposed a new detection algorithm for the closest point on a curve to a point on a screen: As for the applications, this paper also proposed an interactive deformation of human bodies.

The conclusion is described as follows.

(1) We optimize the metaball parameters by the Steepest Descent method to generate the body model based on the measured data obtained from range data.

(2) We propose a method of free-form deformation using Bezier curves. We can get the closest point on a curve using the Bezier Clipping method, which uses the convex hull property of Bezier curves and is an iterative method using linear equations with a small number of iterations. This method is useful for interactive operations such as deformation because of its quick calculation.

(3) The metaball human body models used for cloth simulation are displayed as excellent examples to illustrate the proposed method.

Acknowledgements

We thank Kazuhisa Takae for his technical help in making human body model.

References

[1] T. Nishita, T. Sederberg, M. Kakimoto, "Ray Tracing Trimmed Rational Surface Patches", Computer Graphics, Vol.24, No.4, 1990-8, pp.337-345.

[2] L. Piegle, W. Tiller, The NURBS Book, 1995, pp.229-234.

[3] H. Nishimura, M. Hirai, T. Kawai, T. Kawada, I. Shirakawa, and K. Omura, "Object Modeling by Distribution Function and a Method of Image Generation", Denshi Tsushin Gakkai Ronbunshi, Vol.J68-D, No.4, 1985, pp.718-725 (in Japanese).

[4] S. Muraki, "Volumetric Shape Description of Range Data using Blobby Model", Computer Graphics, Vol.25, No.4, 1991, pp. 227-23.

[5] E. Bittar, N. Tsingos, M. Gascuel, "Automatic Reconstruction of Unstructed 3D Data:Combining a Medial Axis and Implicit Surfaces", EUROGRAPHICS'95, Vol.14, No.3, 1995, pp. 457-468.

[6] T. Beier, S. Neely, "Feature-Based Image Metamorphosis", Computer Graphics, Vol.26, No.2 , 1992, pp.35-42.

[7] G. Wolberg, "Digital Image Warping", IEEE Computer Society Press, 1990.

[8] T. Nishita, K. Fujii, E. Nakamae, "Metamorphosis using Bezier Clipping", Pacific Graphics'93, 1993, pp.162-173.

[9] T. Sederberg, S. Parry, "Free-Form Deformation of Solid Geometric Models", Computer Graphics, Vol.20, No.4 , 1986, pp.151-160.

[10] H. Okabe, H. Imaoka, T. Tomiha, and H. Niwaya, "3 Dimensional Apparel CAD System", Computer Graphics, Vol.26, No.2, 1992, pp.105-110.

[11] R. Matsuda and H. Imaoka, "Modeling of Human Body by Density Balls for Garment Design", Sen'i Gakkaishi, Vol.50, No.5, 1994, pp.221-228 (in Japanese).

[12] R.Matsuda and H.Imaoka, "A Graphic Method to Simulate Grament Fitting on a Human Body Model in Various Posture", Sen'i Gakkaishi, Vol. 51, No.5, 1995, pp.225-233 (in Japanese).