

**On Extracting the Wire Curves from Multiple Face Models
for Facial Animation**

Hyewon Pyun Hyun Joon Shin Sung Yong Shin

CS/TR-165

August 2001

**K A I S T
Department of Computer Science**

On Extracting the Wire Curves from Multiple Face Models for Facial Animation

Hyewon Pyun Hyun Joon Shin Sung Yong Shin

Abstract

Wire curve [15] is a simple, intuitive interface to local deformation of complex geometric objects such as human face models. In this paper, we provide a formulation to extract wire curves and deformation parameters from a facial model based on the displacements of its vertices from those of the corresponding reference model. This extraction process is an inverse process of the wire deformation. With a mild assumption and interactive guide for setting the reference curves and their attributes, we show that the inverse process can be nicely formulated as an over-constrained system of linear equations that can be solved with a least squares minimization technique. We apply the extraction process to multiple face models with different types of expressions to obtain their corresponding wire curves. For facial animation, we blend those extracted wire curves and deformation parameters to finally deform the reference face model. Our proposed scheme facilitates both local deformation and non-uniform blending by making use of the power of wire deformation.

Keywords: real-time facial animation, multiple face models, local deformation, wire deformation, face pattern recognition.

1 Introduction

Multiple face models, called templates, are widely used for facial animation[5, 12]. Each of these models reflects both a facial expression of different type and designer’s insight to be a good guideline for animation. Those templates comprise a facial expression database from which we select appropriate face models to blend and to deform. However, a template consists of a set of vertices with few handles to control such geometric operations except the vertices themselves. To achieve smooth local deformation and non-uniform blending without any tools, one would have to deal with every vertex of face models involved in these operations.

Due to its capability of local control on facial features such as eyes and a mouth, “wires”[15] have become a popular tool for facial animation. The basic idea of wire deformation is to locally deform the vertices of a facial feature according to the displacements of wire curves from their references together with deformation parameters (see Section 2.) Therefore, by deforming a wire curve, the features near the curve are also deformed accordingly. These features may be refined further with deformation parameters. Recently, wire deformation has been incorporated in a well-known animation software called MayaTM as a standard deformation tool.

However, with a variety of modeling tools available, we cannot expect that a designer does necessarily employ wire deformation to model a template of facial expression. Moreover, even modeled with the wire deformation tool, the final result is represented not by wire curves and deformation parameters applied to a reference face model, but by the vertices themselves displaced by those curves and parameters. In order to facilitate the local control of facial features, we present a method to extract a set of wire curves and deformation parameters from a template regardless of its history, that is, how

it has been created. This method not only provides handles for local deformation and non-uniform blending but also reduces the volume of the template database, by representing a template as a set of wire curves and deformation parameters characterizing its corresponding facial expression.

The remainder of the paper is organized as follows. We provide related work in Section 2. In Section 3, we give an introduction to the wire deformation technique. We present a formulation for extracting wire curves and deformation parameters in Section 4. In Section 5, we demonstrate how our technique can be used for facial animation. Finally, we conclude this paper in Section 6.

2 Related Work

Blending multiple face models with different expressions is popular for real-time facial animation. Pighin et al. [12] captured face geometry and textures by fitting a generic face model to a number of photographs. Through transitions between captured face models of different expressions, they were able to generate expressive facial animation. Blanz et al. [1] proposed an automatic face modeling technique by linearly blending a set of example face models from a face database. To avoid unlikely faces, they restricted the range of allowable faces with constraints derived from the example set. For local control on each facial feature, those approaches allow interactive segmentation of a face into a set of regions to assign a proper blending factor to every vertex.

Other alternatives are based on deformation techniques. In free-form deformation(FFD) [14], control points of a parallelepiped-shaped lattice are manipulated to deform an object. Further extensions to FFD adopted lattices of arbitrary topology instead of regular lattices [2, 3, 10]. For direct control, Hsu et al. [6] computed the displacements of the control points from the movements of points on the surface of an object. Thalmann et al. [7] employed FFD to simulate the muscle action on the skin surface of a human face. Terzopoulos et al. [9] proposed a physically-based method for skin and muscle deformation to enhance the degree of realism over purely geometric techniques. Williams [16] and Guenter et. al. [4] used facial motion data captured from real actors to deform face models. Marschner et al. [11] computed the displacements of control points for a specific face model from the movements of sample points on a face performer by solving a system of linear equations in the least squares sense.

Singh et al. [15] provided a more effective control metaphor based on wire deformation. A parametric curve called "wire curve" is used to define and directly control a salient deformable feature on a face model. Its basic idea is to locally deform geometry near the wire curve by manipulating the curve. Due to the capability of local control as well as direct manipulation, wire deformation is versatile for synthesizing facial expressions interactively.

3 Wire Deformation

Singh *et. al.* [15] proposed wire deformation as a simple, intuitive interface to deforming complex geometric objects such as human face models. In this section, we briefly summarize their deformation scheme.

Wire deformation is defined by a tuple $\langle W, R, f, r, s \rangle$, where W and R denote parametric curves called wire and reference curves, respectively, and f , r , and s are deformation parameters to be explained later. Initially, W and R are coincident. By deforming the wire curve W , it is displaced from R . For a point \mathbf{p} on an object M to deform, let \mathbf{p}_R be its nearest point on R , and \mathbf{p}_W the point on W corresponding to \mathbf{p}_R . That is, \mathbf{p}_R and \mathbf{p}_W have the same curve parameter value. When W is

deformed, the point \mathbf{p} is moved to \mathbf{p}' as follows.

$$\mathbf{p}' = \mathbf{p} + (\mathbf{p}_W - \mathbf{p}_R)\mathbf{f}(\mathbf{x}). \quad (1)$$

Here, x is a function of R , \mathbf{p} , and the range parameter r , that is,

$$x = x(R, \mathbf{p}, \mathbf{r}).$$

x is proportional to the Euclidean distance from \mathbf{p}_R to \mathbf{p} , that is normalized by r . In particular,

$$x = \frac{\|\mathbf{p} - \mathbf{p}_R\|}{r}.$$

The function f in Equation (1) is a monotonically decreasing function of x that satisfies $f(0) = 1$ and $f(x) = 0$ for $x \geq 1$. We use

$$f(x) = \begin{cases} (x^2 - 1)^2 & , \text{ if } 0 \leq x \leq 1, \\ 0 & , \text{ otherwise.} \end{cases}$$

f gives the fraction of displacement $(\mathbf{p}_W - \mathbf{p}_R)$ to apply to \mathbf{p} to obtain \mathbf{p}' . The distance $\|\mathbf{p} - \mathbf{p}_R\|$ and the range parameter r determine x uniquely and thus f . We can also adjust $f(x)$ interactively within the unit interval $[0, 1]$ to reflect the physical characteristics of the point \mathbf{p} in relation to R . With r fixed, f is decreasing as \mathbf{p} is farther away from R . The wire deformation in Equation (1) can be enhanced further with a radial scaling parameter s , that is,

$$\mathbf{p}' = \mathbf{p} + (s - 1)(\mathbf{p} - \mathbf{p}_R)\mathbf{f}(\mathbf{x}) + (\mathbf{p}_W - \mathbf{p}_R)\mathbf{f}(\mathbf{x}). \quad (2)$$

The scaling parameter s controls the movement of \mathbf{p} in the direction of $\mathbf{p} - \mathbf{p}_R$ or its reverse. The second term of the right-hand side of Equation (2) is for scaling. To assure no scaling when $s = 1$, we have modified the original work. We also ignore the rotation term since rotations are rarely employed for deformation of face models.

Multiple wire curves can be used to better control the shape of the object. In their original work, Singh *et. al.* proposed three alternatives to assign a weight to the contribution of each wire curve to the final deformation. For our purpose, we choose the one described below: Let $\Delta \mathbf{p}_j$ be the displacement of the point \mathbf{p} when a wire curve W_j alone is applied. Given n wire curves W_j , $j = 0, 1, 2, \dots, m$, the new position \mathbf{p}' is obtained as follows:

$$\mathbf{p}' = \mathbf{p} + \frac{\sum_{j=0}^m \Delta \mathbf{p}_j f_j(\mathbf{x})^k}{\sum_{j=0}^m f_j(\mathbf{x})^k}. \quad (3)$$

Here, $f_j(x) = f(x(R_j, \mathbf{p}, \mathbf{r}_j))$ where R_j and r_j are the reference curve and the range parameter corresponding to W_j . The localizing parameter k controls the influence of W_j and s_j on deformation. For example, as $f_j(x)$ approaches one (or \mathbf{p} approaches R_j), the influence of W_j and s_j are more rapidly increasing with bigger k .

4 Wire Extraction

Suppose that we use m pairs of wire and reference curves, denoted by (W_j, R_j) , $j = 0, 1, 2, \dots, m$, to characterize the geometry of a facial expression template T deformed from the base model M by displacing their vertex positions. Then, our problem is:

Given M and T , determine the curve pairs, (W_j, R_j) , $j = 0, 1, 2, \dots, m$ and deformation parameters, f , r , s , and k such that T can be obtained from M through wire deformation by using those curve pairs and parameters.

The reference curves $R_j(u)$, $j = 0, 1, 2, \dots, m$ characterize the features of the base face model M such as eyes and the mouth and thus are not dependent on a specific template T . The range parameter $r_j(u)$ is an attribute of the curve $R_j(u)$ defined along it for all u . The localizing parameter k is applied uniformly to every point on M regardless of T . Therefore, we assume that experienced designers specify R_j 's, r_j 's, and k interactively to reflect their intuition on M . Provided with R_j 's and r_j 's, $f(x(R_j, \mathbf{p}, \mathbf{r}_j))$ is uniquely computed. Hence, we will be done if the radial scaling parameters s_j 's and the wire curves W_j 's are determined. Here, s_j is a useful parameter for expression exaggeration or attenuation. The cheek bulge for an exaggerated smile is a good example of a very much use of s_j .

Let \mathbf{p}_i and \mathbf{p}'_i , $i = 0, 1, 2, \dots, n$ be the vertices of M and its corresponding vertices of T . From equation (3),

$$\Delta \mathbf{p}_i = \frac{\sum_{j=0}^m f_{ij}(x)^k \Delta \mathbf{p}_{ij}}{\sum_{j=0}^m f_{ij}(x)^k}, i = 0, 1, 2, \dots, n. \quad (4)$$

Here, $\Delta \mathbf{p}_{ij}$ is the displacement of \mathbf{p}_i when only W_j is applied, $f_{ij} = f(x(R_j, \mathbf{p}_i, \mathbf{r}_j))$, and $\Delta \mathbf{p}_i = \mathbf{p}'_i - \mathbf{p}_i$. From Equation (2),

$$\Delta \mathbf{p}_{ij} = (s_j - 1)f_{ij}(x)(\mathbf{p}_i - \mathbf{p}_{iR_j}) + f_{ij}(x)(\mathbf{p}_{iW_j} - \mathbf{p}_{iR_j}), \quad (5)$$

where \mathbf{p}_i , s_j , \mathbf{p}_{iR_j} , and \mathbf{p}_{iW_j} are a vertex \mathbf{p}_i on M , the scaling parameter of the reference curve R_j , the point on R_j closest to \mathbf{p}_i , and the point on W_j corresponding to \mathbf{p}_{iR_j} . As stated in the previous section, W_j and R_j are parametric curves. In particular, we employ cubic B-splines to represent them. Therefore, $(\mathbf{p}_{iW_j} - \mathbf{p}_{iR_j})$ can be expressed as follows:

$$(\mathbf{p}_{iW_j} - \mathbf{p}_{iR_j}) = \sum_{l=0}^{t_j} \mathbf{B}_l(\mathbf{w}_{jl} - \mathbf{r}_{jl}), \quad (6)$$

where \mathbf{w}_{jl} and \mathbf{r}_{jl} , $l = 0, 1, 2, \dots, t_j$ are the control points of wire and reference curves W_j and R_j , respectively, and B_l , $l = 0, 1, 2, \dots, t_j$ are their basis functions.

From Equations (5) and (6),

$$\Delta \mathbf{p}_{ij} = (s_j - 1)f_{ij}(x)(\mathbf{p}_i - \mathbf{p}_{iR_j}) + f_{ij}(x) \sum_{l=0}^{t_j} \mathbf{B}_l(\mathbf{w}_{jl} - \mathbf{r}_{jl}). \quad (7)$$

Given \mathbf{p}_i , we can determine \mathbf{p}_{iR_j} and its curve parameter value on R_j and thus B_l can be evaluated. The only unknowns on the right-hand side of Equation (7) is the control points \mathbf{w}_{jl} 's of the wire curve W_j and their scaling factor s_j 's. From Equations (4) and (7),

$$\Delta \mathbf{p}_i = \sum_{j=0}^m \frac{f_{ij}(x)^k}{\sum_{j=0}^m f_{ij}(x)^k} [(s_j - 1)f_{ij}(x)(\mathbf{p}_i - \mathbf{p}_{iR_j}) + f_{ij}(x) \sum_{l=0}^{t_j} \mathbf{B}_l(\mathbf{w}_{jl} - \mathbf{r}_{jl})]. \quad (8)$$

We are going to solve Equation (8) for s_j and W_j . In a special case, we can trivially get W_j . Suppose that R_j passes through a sequence of vertices on the base model M and also that only one wire curve is defined on M . Then, R_j and W_j become R_0 and W_0 , respectively, since we have one wire curve. For vertex \mathbf{p}_i on R_0 , the first term of the equation vanishes since $\mathbf{p}_i - \mathbf{p}_{iR_0} = \mathbf{0}$. Moreover,

$f_{i0}(x) = 1$ since the vertex \mathbf{p}_i is on R_0 . Therefore, Equation (8) is reduced to $\Delta\mathbf{p}_i = \mathbf{p}_i\mathbf{w}_0 - \mathbf{p}_i\mathbf{R}_0$. That is, the displacement of vertex \mathbf{p}_i is determined only by a single wire curve W_0 . Therefore, W_0 can be computed from the vertices on the expression template T corresponding to those on M . However, multiple wire curves are generally defined on M for facial animation. The vertex \mathbf{p}_i on M moves to a new position by the influence of multiple wire curves. Hence, even if R_j passes through the sequence of vertices on M , W_j can not be obtained from the same sequence of vertices on T corresponding to those on M .

Let

$$\begin{aligned} w_{ij} &= (f_{ij}(x)^{k+1}) / (\sum_{j=0}^m f_{ij}(x)^k), \\ \mathbf{c}_i &= \sum_{j=0}^m w_{ij}(\mathbf{p}_i - \mathbf{p}_{iR_j}), \text{ and} \\ \mathbf{q}_{jl} &= \mathbf{w}_{jl} - \mathbf{r}_{jl}, l = 0, 1, 2, \dots, t_j. \end{aligned}$$

Then, Equation (8) becomes

$$\Delta\mathbf{p}_i = \mathbf{c}_i + \sum_{j=0}^m \mathbf{w}_{ij}((\mathbf{p}_i - \mathbf{p}_{iR_j})\mathbf{s}_j + \sum_{l=0}^{t_j} \mathbf{B}_l \mathbf{q}_{jl}), i = 0, 1, 2, \dots, n. \quad (9)$$

Here, \mathbf{q}_{jl} 's and s_j 's are the only unknowns. Rearranging Equation (9), we have a system of linear equations:

$$\sum_{j=0}^m w_{ij}(\mathbf{p}_i - \mathbf{p}_{iR_j})\mathbf{s}_j + \sum_{j=0}^m \sum_{l=0}^{t_j} \mathbf{w}_{ij}\mathbf{B}_l \mathbf{q}_{jl} = \Delta\mathbf{p}_i - \mathbf{c}_i, i = 0, 1, 2, \dots, n, \quad (10)$$

or

$$\mathbf{Bs} + \mathbf{Cq} = \mathbf{b}, \quad (11)$$

where

$$\begin{aligned} \mathbf{B} &= \begin{bmatrix} w_{00}(\mathbf{p}_0 - \mathbf{p}_{0R_1}) & w_{01}(\mathbf{p}_0 - \mathbf{p}_{0R_2}) & \cdots & w_{0m}(\mathbf{p}_0 - \mathbf{p}_{0R_m}) \\ w_{10}(\mathbf{p}_1 - \mathbf{p}_{1R_1}) & w_{11}(\mathbf{p}_1 - \mathbf{p}_{1R_2}) & \cdots & w_{1m}(\mathbf{p}_1 - \mathbf{p}_{1R_m}) \\ \vdots & \vdots & \ddots & \vdots \\ w_{n0}(\mathbf{p}_n - \mathbf{p}_{nR_1}) & w_{n1}(\mathbf{p}_n - \mathbf{p}_{nR_2}) & \cdots & w_{nm}(\mathbf{p}_n - \mathbf{p}_{nR_m}) \end{bmatrix}, \\ \mathbf{C} &= \begin{bmatrix} w_{00}B_0 & w_{00}B_1 & \cdots & w_{00}B_{t_0} & w_{01}B_0 & w_{01}B_1 & \cdots & w_{01}B_{t_1} & \cdots & w_{0m}B_0 & w_{0m}B_1 & \cdots & w_{0m}B_{t_m} \\ w_{10}B_0 & w_{10}B_1 & \cdots & w_{10}B_{t_0} & w_{11}B_0 & w_{11}B_1 & \cdots & w_{11}B_{t_1} & \cdots & w_{1m}B_0 & w_{1m}B_1 & \cdots & w_{1m}B_{t_m} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ w_{n0}B_0 & w_{n0}B_1 & \cdots & w_{n0}B_{t_0} & w_{n1}B_0 & w_{n1}B_1 & \cdots & w_{n1}B_{t_1} & \cdots & w_{nm}B_0 & w_{nm}B_1 & \cdots & w_{nm}B_{t_m} \end{bmatrix}, \\ \mathbf{q} &= (\mathbf{q}_{00} \ \mathbf{q}_{01} \ \cdots \ \mathbf{q}_{0t_0} \ \mathbf{q}_{10} \ \mathbf{q}_{11} \ \cdots \ \mathbf{q}_{1t_1} \ \cdots \ \mathbf{q}_{mt_m})^T, \\ \mathbf{s} &= (s_1 \ s_2 \ \cdots \ s_m)^T, \text{ and} \\ \mathbf{b} &= (\Delta\mathbf{p}_0 - \mathbf{c}_0 \ \Delta\mathbf{p}_1 - \mathbf{c}_1 \ \cdots \ \Delta\mathbf{p}_n - \mathbf{c}_n)^T. \end{aligned}$$

The vector \mathbf{s} represents the radial scaling factors. Each element \mathbf{q}_{il} of the vector \mathbf{q} is the displacement of the control point \mathbf{w}_{il} from \mathbf{r}_{il} . We can further simplify Equation (11), juxtaposing matrices \mathbf{B} and \mathbf{C} :

$$\mathbf{A}\hat{\mathbf{q}} = \mathbf{b}, \quad (12)$$

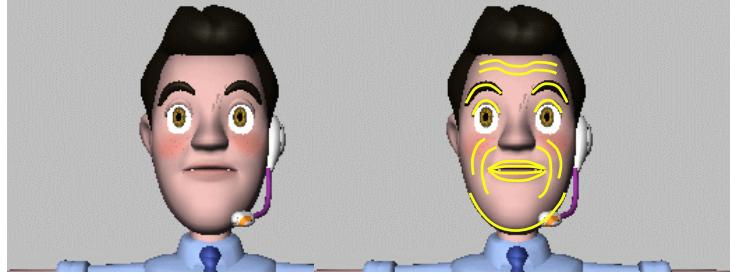


Figure 1: The base facial model and its corresponding wire curves.

where $\mathbf{A} = [\mathbf{B} | \mathbf{C}]$ and $\hat{\mathbf{q}} = (\mathbf{s}^T, \mathbf{q}^T)^T$. Solving Equation (12) for $\hat{\mathbf{q}}$, we can not only find the scaling factors but also extract all wire curves. The system given in Equation (12) is over-constrained, since the number of vertices in the face model M is much greater than the total number of control points for all reference (or equivalently wire) curves and their scaling factors. Therefore, we need to compute the least squares solution, that is,

$$\hat{\mathbf{q}} = (\mathbf{A}^T \mathbf{A})^+ \mathbf{A}^T \mathbf{b}. \quad (13)$$

$(\mathbf{A}^T \mathbf{A})^+$ is the pseudo inverse of $(\mathbf{A}^T \mathbf{A})$ obtained from its singular value decomposition [8, 13]. The first m elements of $\hat{\mathbf{q}}$ give the scaling factors s_j 's for the template T . Displacing the control points of the reference curves (or equivalently the initial wire curves) with the rest of elements, we finally compute the control points of all wire curves.

5 Experimental Results

For our experiments, we have built a base face model and its templates of different expression types. The base model consists of about 5,000 polygons. The expression templates are derived by designers from the base model through displacing the vertices of the base model. As shown in figure 1, we define 15 reference (and thus wire) curves lying on the base model. Each of reference and wire curves is a cubic B-spline and has four or more control points.

First, we show how well our wire extraction scheme works. In Figure 2, the original face templates are arranged side by side with their corresponding templates reconstructed from the base face model by applying the extracted wire curves and deformation parameters. In the first column, we give the original template that have sad, happy, surprised and angry expressions from top to bottom, respectively. In the second column, we show the corresponding reconstructed templates, in the same sequence. Note how visually similar the pairs of original and reconstructed expressions are. This supports the effectiveness of our scheme to extract the wire curves and deformation parameters.

Now, we exhibit the capability of wire curves for local deformation. With the wire curves and deformation parameters extracted, we can use them as a high level user interface to locally deform facial features such as eyes, lips, forehead, cheeks, and etc, instead of interactively manipulating every vertex involved in the deformation, individually. Figure 3 shows local deformation achieved with such wire curves: The left figure shows a smiling expression and its corresponding wire curve configuration, the expression in the middle is obtained by manipulating mainly the wire curves that characterize the lips, and the expression on the right is obtained by deforming mainly the wire curves on eyebrows.

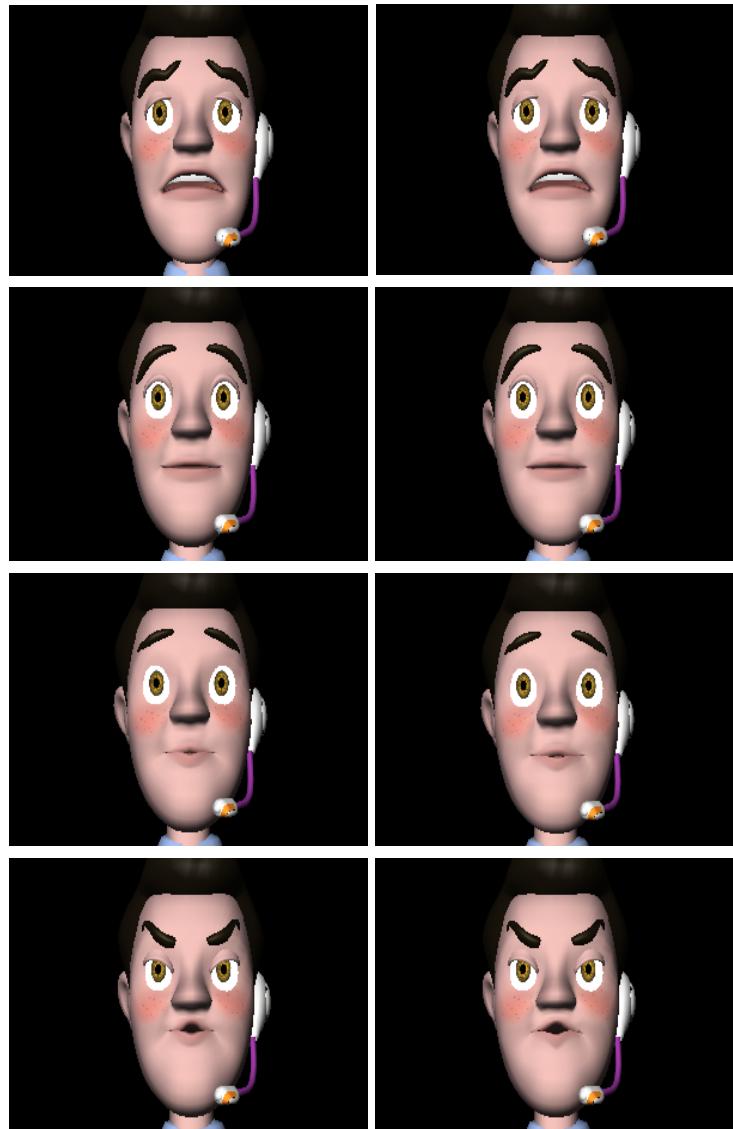


Figure 2: The original face templates and their corresponding reconstructed models.

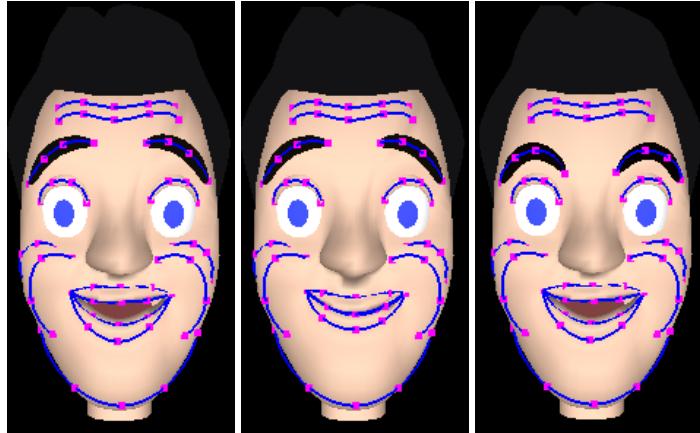


Figure 3: The examples of high level user interface for local deformation.

Finally, we demonstrate the ease of non-uniform blending with wire curves extracted. The upper row of Figure 4 shows the uniform blending of facial features, that is, uniform feature interpolation between two templates represented by image (1) and image (10), respectively. Each facial feature of the former is transited to that of the latter at the same speed. The lower row gives an image sequence due to their feature-wise non-uniform blending. For this non-uniform blending, we use different blending functions for eyes and the mouth. As shown in Figure 6, the blending function for eyes is $g(t) = \sqrt{t}$, and the blending function for the mouth is $g(t) = t^3$ for $0 \leq t \leq 1$. These blending functions enable the transition of eye features to be much faster than that of lip features when t is near zero. On the other hand, the latter is much faster than the former when t approaches one. Feature-wise non-uniform blending can hardly be achieved efficiently without an effective user-interface such as the wire deformation scheme. Figure 5 shows an example of uniform and non-uniform blending for another model.

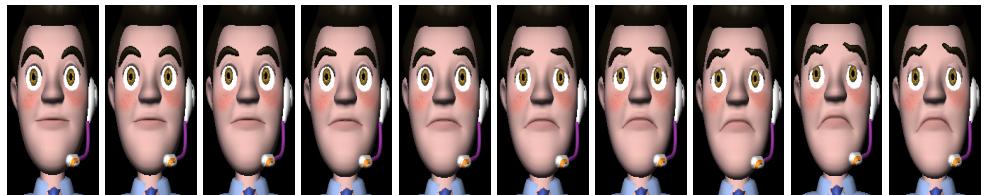
6 Conclusion

In this paper, we present a method to extract a set of wire curves and deformation parameters from a face model regardless of its construction history. Given a pair of reference and face models with an identical topological structure, we formulate a system of linear equations of which the unknowns are the positions of control points of each wire curve and scaling parameters. This system is over-constrained since the number of vertices in the face models is much greater than that of unknowns. We extract the wire curves and parameters by solving the system for the least squares solution. The wire curves together with the scaling parameters, thus extracted, not only provide convenient handles for local geometry control but also facilitate non-uniform transitions among facial expression templates. The experimental results show the effectiveness of wire curve extraction and its usefulness.

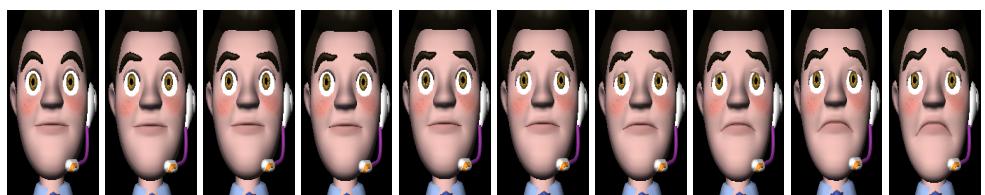
In the future, we plan to extract the positions of reference curves and more deformation parameters. With the initial position of a reference curve interactively given by a designer, we will try to extract the best position of the reference curve that fits for a facial expression model and its reference.

References

- [1] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. *SIGGRAPH 1999 Conference Proceedings*, pages 187–194, 1999.
- [2] Y. K. Chang and A. P. Rockwood. A generalized de casteljau approach to 3d free-form deformation. *SIGGRAPH 94*, 28:257–260, 1994.
- [3] Sabine Coquillart. Extended free-form deformation: A sculpturing tool for 3d geometric modeling. *SIGGRAPH 90*, 24:187–196, 1990.
- [4] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Frederic Pighin. Making faces. *SIGGRAPH 98 Conference Proceedings*, pages 55–67, 1998.
- [5] Adele Hars. Masters of Motion Capture. *Computer Graphics World*, pages 27–34, October 1996.
- [6] William M. Hsu, John F. Hugues, and Henry Kaufman. Direct manipulation of free-form deformation. *SIGGRAPH 92*, pages 177–184, 1992.
- [7] P. Kalra, A. Mangili, N. M. Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. *Eurographics 92*, 58:59–69, 1992.
- [8] Min-Ho Kyung, Myung-Soo Kim, and Sung Je Hong. A New Approach to Through-the-Lens Camera Control. *CVGIP: Graphical Models and Image Processing*, 58(3):262–285, 1996.
- [9] Yuencheng Lee, D. Terzopoulos, and K. Waters. Realistic modeling for facial animation. *SIGGRAPH 95 Conference Proceedings*, pages 55–62, 1995.
- [10] R. MacCracken and K. Joy. Free-form deformations with lattices of arbitrary topology. *SIGGRAPH 96*, 30:181–189, 1996.
- [11] Stephen R. Marschner, Brian Guenter, and Sashi Raghupathy. Modeling and rendering for realistic facial animation. *EUROGRAPHICS Rendering Workshop 2000*, pages 98–110, 2000.
- [12] Frederic Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, and David H. Salesin. Synthesizing Realistic Facial Expressions from Photographs. *SIGGRAPH 98 Conference Proceedings*, pages 75–84, 1998.
- [13] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes in C: Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.
- [14] Sederberg, Thomas W., Parry, and Scott R. Free-form deformation of solid geometric models. *SIGGRAPH 86*, 20:151–160, 1986.
- [15] Karan Singh and Eugene Fiume. Wires: A Geometric Deformation Technique. *SIGGRAPH 98 Conference Proceedings*, pages 299–308, 1998.
- [16] Lance Williams. Performance-driven facial animation. *SIGGRAPH 90*, 24:235–242, 1990.



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

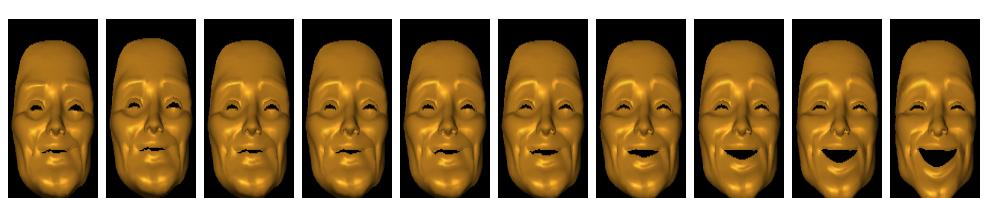


(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

Figure 4: upper row: uniform blending. lower row: non-uniform blending.



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)



(1) (2) (3) (4) (5) (6) (7) (8) (9) (10)

Figure 5: upper row: uniform blending. lower row: non-uniform blending.

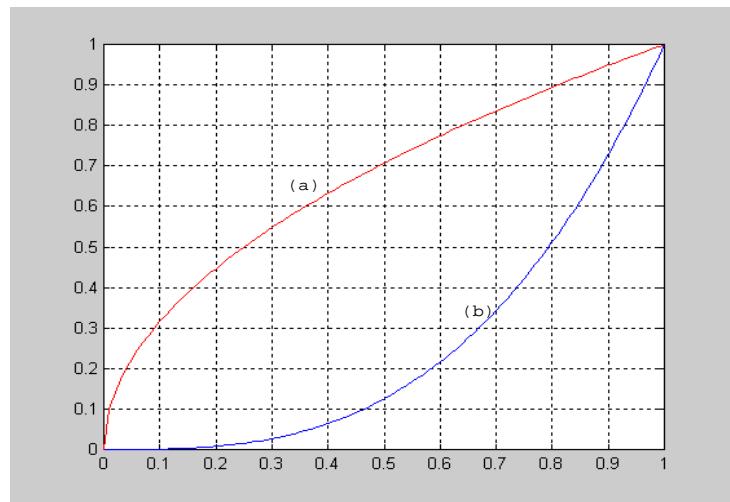


Figure 6: (a) The blending function for eyes, $y = \sqrt{t}$ for $0 \leq t \leq 1$. (b) The blending function for mouth $y = t^3$ for $0 \leq t \leq 1$.