Computer Integrated Manufacturing Systems Vol. 11, No. 4, p. 253–265, 1998

Printed in Great Britain
0951-5240/98/$ — see front matter
ELSEVIER
PII: S0951-5240(98)00023-8

Modification of a NURBS curve with nose features

Wei-Cherng Lu and Jiung-Ming Huang

Department of Mechanical Engineering, National Taiwan University of Science and Technology, 43, Keelung Road, Section 4, Taipei, Taiwan 106

This paper proposes a methodology to modify the shape of a NURBS curve by moving the nose points of each curve segment. To perform the modification, the algorithms of evaluating control point movements and weight changes are first introduced. In weight changes, the derived equation for modifying a curve segment is viewed as an equilibrium force system, acting on the target nose point and yielding the required nose point displacement, which provides the foundation for the algorithm to attain the required curve modification. To raise the precision and efficiency of curve modification, a method of constraining the joining points and the nose points is proposed. In addition, a method is presented to reach an optimal solution to curve modification, and is verified in the implementation.

Keywords: NURBS curve, nose feature, shape modification

Introduction

Shape design and modification for free-form geometry is a typical task in the automobiles, aircraft, molds, and artworks industries. It begins with extracting the free-form features, and next defines the 'features' frameworks either by sketching or through a data set of CMM measurement. A basic CAD model is then created by editing and relating the constituent features[1]. Finally, a desired shape modification is accomplished after interactive change of geometric features.

A NURBS (Non-Uniform Rational B-Spline) curve is a vector-valued piecewise rational function, which is widely adopted in the CAD/CAM and computer graphics community. A NURBS curve, mathematically speaking, avails to some parametric manipulation for its modifications: moving control points, elevating the order of blending functions, inserting knots, modifying tangent magnitudes or curvatures, and changing weights through rational polynomials, etc.[2].

There are several methods found in the literature to modify a NURBS curve, as listed below:

1. Translate the curve segments in a certain direction[3–6]: This method aims to move the control points to modify the curve shape. As *Figure 1* shows, when a control point moves, the affected curve segment will move in the same direction as the control point does.
2. Pull/push the curve segments[3,5]: This approach uses the rational curve model to flatten or sharpen segment noses by adjusting the weights of control points. As illustrated in *Figure 2*, when the weight of a control point changes, the affected curve

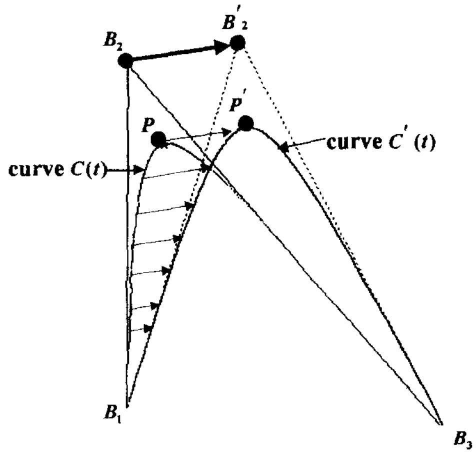

Figure 1 A NURBS curve modification by moving the nose control point.

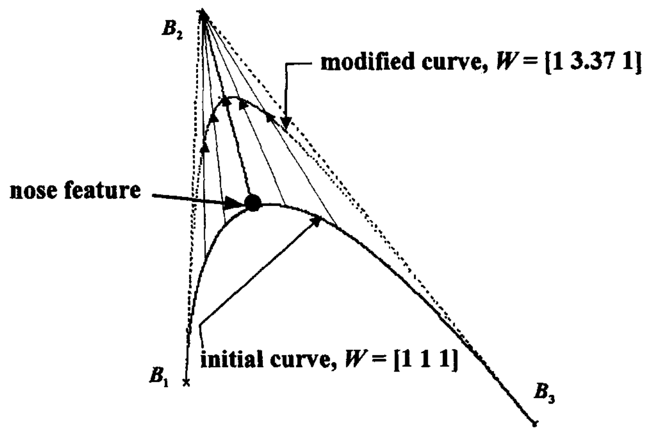

Figure 2 A NURBS curve modification by changing the weight for the nose control point $\mathbf{B}_2$.

segment will move toward (or away from) the nose control point of this curve segment.

3. Modulate the knot positions and length ratio of the curve segments[7]: In this approach, knot spans are reallocated to modify the knot values for joining points between curve segments. *Figure 3* shows the results using different knot vectors.
4. Subdivide segments, elevate and reduce the degree[8-10]: These method focus on inserting knots or changing the order of the defining B-spline basis functions to modify or design a curve.

For the issue of curve modification, the movement of feature points on each curve segment toward target positions is crucial to modify the shape features of an artistic curve. There are, however, few articles which discuss the methodology to modify or design a curve by changing the curve features. To achieve the goal of curve modification, it is advisable to firstly well divide a curve into several featured segments, and then modify the curve shape by merely moving the features. This paper thus aims to propose a methodology which functions to modify a NURBS curve based on constraints of the so-called 'nose points' of each curve segment. A scheme is laid out to discover new control points and weights for curve modification based on the nose points. When the

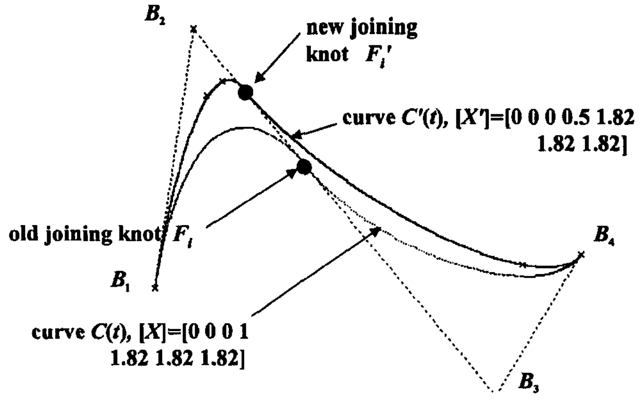

Figure 3 A NURBS curve modification by changing the knot vector.

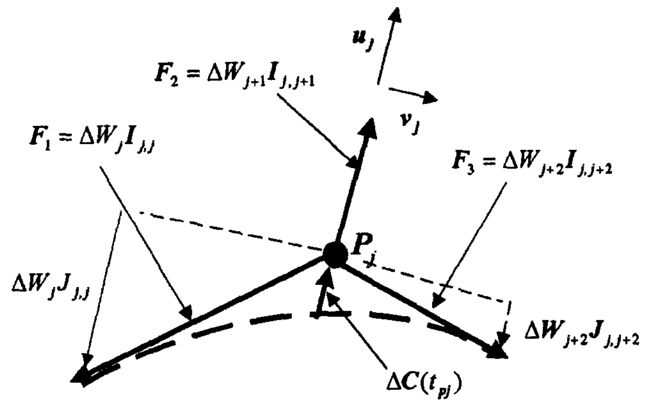

Figure 4 A curve modification with multiple noses.

nose points are close to their target positions by using new weights or control points, other points of the curve segments may more or less deflect. To moderate the deflection, simply changing the knot vector can achieve this demand. Also discussed in this paper is a set of formulae for calculating new knot spans to force the joining points to target positions.

Algorithm for curve modification

Assume a NURBS curve is composed of n curve segments (see *Figure 4*). Let a curve point on curve segment j be $C(t_{pj})$ and its target position P_j, where t_{pj} represents the parameter value of point $C(t_{pj})$. Express these two points by the following NURBS form[7]:

$$\frac{\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_iB_i}{\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i} - \frac{\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_iB_i}{\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i} = P_j - C(t_{pj}) \quad (1)$$

where W_i and W_i' express the weights for control points B_i before and after curve modification, $N_{i,k}$ is the basis function of order k, and B_i and B_i' are the control points before and after curve modification.

Let $W_i' = W_i + \Delta W_i, B_i' = B_i + \Delta B_i$ and rearrange the above equation into the following form:

$$\begin{aligned}
&\Sigma N_{i,k}(t_{pj})W_i\Sigma N_{i,k}(t_{pj})(W_i+\Delta W_i)\Delta B_i \\
&\quad + \Sigma N_{i,k}(t_{pj})W_i\Sigma N_{i,k}(t_{pj})W_iB_i \\
&\quad + \Sigma N_{i,k}(t_{pj})W_i\Sigma N_{i,k}(t_{pj})\Delta W_iB_i \\
&\quad - \Sigma N_{i,k}(t_{pj})W_i\Sigma N_{i,k}(t_{pj})W_iB_i \\
&\quad - \Sigma N_{i,k}(t_{pj})\Delta W_i\Sigma N_{i,k}(t_{pj})W_iB_i \\
&= \Sigma N_{i,k}(t_{pj})W_i[\Sigma N_{i,k}(t_{pj})W_i(P_j - C(t_{pj})) \\
&\quad + \Sigma N_{i,k}(t_{pj})\Delta W_i(P_j - C(t_{pj}))]
\end{aligned}$$

Multiply the above equation by

$$\frac{1}{\Sigma N_{i,k}(t_{pj})W_i}$$

to give

$$\Sigma N_{i,k}(t_{pj})(W_i+\Delta W_i)\Delta B_i + \Sigma N_{i,k}(t_{pj})\Delta W_iB_i$$

$$-\Sigma N_{i,k}(t_{pj})\Delta W_i \frac{\Sigma N_{i,k}(t_{pj})W_iB_i}{\Sigma N_{i,k}(t_{pj})W_i}$$
$$=\Sigma N_{i,k}(t_{pj})W_i(P_j-C(t_{pj}))$$
$$+\Sigma N_{i,k}(t_{pj})\Delta W_i(P_j-C(t_{pj}))$$

Since

$$C(t_{pj})=\frac{\Sigma N_{i,k}(t_{pj})W_iB_i}{\Sigma N_{(i,k)}(t_{pj})W_i},$$

the above equation becomes

$$\Sigma N_{i,k}(t_{pj})(W_i+\Delta W_i)\Delta B_i+\Sigma N_{i,k}(t_{pj})\Delta W_i(B_i-P_j)$$
$$=\Sigma N_{i,k}(t_{pj})W_i(P_j-C(t_{pj})) \quad (2)$$

Equation (2) expresses the weight changes ΔW_i and control point movements ΔB_i to move a curve point from curve point $C(t_{pj})$ toward point P_j.

Nose points

In this paper, it is assumed that there is only a single curve point $\Delta C(t_{pj})$ on each curve segment to be moved for modification, and the point P_j can be viewed as the target 'nose point' of this segment. Theoretically speaking, a nose point can be any point on a NURBS curve segment mathematically. Due to the following reasons, it should be specific:

- In practical use, it is very unnatural to select one point for each curve segment on a CAD screen.
- To improve the precision of a fitted curve, selecting some key points from the curve to fit to the target places is more reachable and effective than to fit the curve to all the data set.

However, how do we determine the key points? Prominent curve points, e.g. the sharp points of the convex or concave curve segments, are what we consider the key points for the reasons given below:

- These points almost locate at the farthest positions of the curve from each other. Modify the curve by constraining these points to target positions, the curve segments will change shape more regularly and have little influence on others.
- From *Figure 2*, one can recognize that the points closest to the concave or convex tips of the curve segments are more likely to fit closer to the requirement in equation (7) (give smallest side displacement (v_i-direction) error, and get precise displacements of the curve points).
- These points can be viewed as the feature points on a fitted curve, so that one can improve the curve fitting by moving them to target nose points of the original data set without loosing the smoothness.

In this paper, a nose point refers to the unique point of a curve segment, which is the curve point taken with a parameter value of the average parameter of the two boundary joining points. Alternatively, one may choose the curve point which is nearest to the second governing control point (B2 in *Figure 2*) for this curve segment. Because of the uniqueness of this point, a nose point can thus be named as a 'nose feature' for the curve segment. Before the user can naturally identify and modify these nose points, the nose points $\Delta C(t_{pj})$ should be dotted on the CAD screen in advance.

As to the target nose points, they should be manually input by the user on the CAD screen for curve modification. But for curve-fitting improvement, it is suggested to take the nearest data point to the second governing control points of the curve segment.

The following four sections will discuss the modification of a NURBS curve by the change of weights, the movement of control points, and the positioning of joining knots.

Keep the weights unchanged

The nose point displacements by moving control points are discussed below by viewing both the cases of a single curve segment and multiple curve segments.

Local modification for a single curve segment. If the control points move without any weight change, i.e. $\Delta W_i=0$, equation (2) becomes

$$\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i\Delta B_i=\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i(P_j-C(t_{pj})) \quad (3)$$

For the condition that the jth curve segment is modified by moving the nose control point B_{j+1}, equation (3) can be simplified as:

$$N_{j+1,k}(t_{pj})W_{j+1}\Delta B_{j+1}=\sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i(P_j-C(t_{pj})) \quad (4)$$

Rearrange the above equation and give

$$\Delta B_{j+1}=\frac{\Sigma N_{i,k}(t_{pj})W_i}{N_{j+1,k}(t_{pj})W_{j+1}}(P_j-C(t_{pj})) \quad (5)$$

The above equation indicates the following: if a control point moves, the affected curve points will move in the same direction as that of the control point (but each in different magnitudes). As illustrated in *Figure 1*, it shows that the points on the affected curve segment would translate in direction $\overline{B_2B_2'}$ when a control point B_2 is moved to B_2'.

Modification for multiple curve segments. Suppose two (or more) adjacent curve segments are asked to be modified. The two adjacent control points for the two curve segments can be viewed as the 'nose control points' of the two curve segments. In general, if the nose point on each curve segment is assigned a target position during the modification, equation (3) can be applied for each curve segment. Thus generated are n vector equations with $n+2$ unknowns to move the n nose points for the n curve segments.

Express these n vector equations in x and y directions as:

$$
\begin{aligned}
&b_{1,1}\Delta B_1 + b_{1,2}\Delta B_2 + b_{1,3}\Delta B_3 = a_1 \Delta C(t_{p1})\\
&b_{2,2}\Delta B_2 + b_{2,3}\Delta B_3 + b_{2,4}\Delta B_4 = a_2 \Delta C(t_{p2})\\
&\qquad\vdots\\
&\qquad\vdots\\
&b_{n,n}\Delta B_n + b_{n,n+1}\Delta B_{n+1} + b_{n,n+2}\Delta B_{n+2} = a_n \Delta C(t_{pn})
\end{aligned}
\tag{6}
$$

where $a_j = \Sigma N_{i,3}(t_{pj})W_i, b_{j,i} = N_{i,3}(t_{pj})W_i$

Suppose the end control points are fixed, i.e. $\Delta B_1 = \Delta B_{n+2} = 0$, there will be n equations for n unknowns ΔB_i in equation (6). Equation (6) can be rewritten into the following matrix form:

$$
\begin{bmatrix}
b_{1,2} & b_{1,3} & 0 & & & \cdots & & 0\\
b_{2,2} & b_{2,3} & b_{2,4} & 0 & & & & 0\\
0 & & & 0 & & & & \\
 & 0 & \ddots & & & \ddots & & \vdots\\
\vdots & & \ddots & b_{j,j-1} & b_{j,j} & b_{j,j+1} & & \\
 & & & & & \ddots & & 0\\
 & & & & 0 & b_{n-1,n-1} & b_{n-1,n} & b_{n-1,n+1}\\
0 & & \vdots & & & 0 & b_{n,n} & b_{n,n+1}
\end{bmatrix}
\begin{bmatrix}
\Delta B_2\\ \Delta B_3\\ \\ \\ \\ \Delta B_n\\ \Delta B_{n+1}
\end{bmatrix}
=
\begin{bmatrix}
a_1\Delta C(t_{p1})\\ a_2\Delta C(t_{p2})\\ \vdots\\ a_j\Delta C(t_{pj})\\ \vdots\\ a_n\Delta C(t_{pn})
\end{bmatrix}
\tag{7}
$$

The control point movements ΔB_j will thus be obtained by solving equation (7) in x and y directions, respectively.

Figure 7(a) depicts an initial fitted curve, which can be improved by moving the nose points. The target nose points are the data points calculated nearest to the nose points of the initial fitted curve, and *F*is are the joining points of curve segments. The curve is initially composed of control points **B**: [(124.39, 175.54), (167.68, 286.47), (281.62, 192.61), (407.82, 323.98), (454.80, 163.49), (362.84, 81.97)], weights *W*: [1 1 1 1 1 1], knot vector *X*: [0 0 0 1 1.855 2.843 3.287 3.287 3.287 3.287], and the distance between the initial and target nose points *Err*: [8.963, 5.550, 8.275, 0.396]. Through control point movements by using equation (7), the error vector of nose points becomes *Err*: [0.588, 3.594, 0.299, 1.210] as depicted in *Figure* 7(b).

Keep the control points unmoved

If all the control points remain constant, i.e. $\Delta B_i = 0$, equation (2) becomes

$$
\sum_{i=1}^{n+2} N_{i,k}(t_{pj})\Delta W_i(B_i - P_j) = \sum_{i=1}^{n+2} N_{i,k}(t_{pj})W_i(P_j - C(t_{pj}))
\tag{8}
$$

For a third-order NURBS curve, i.e., $k = 3$, equation (8) can be viewed as the following equilibrium equation, which consists of four concurrent vectors intersecting at the target point P_j.

$$
\begin{aligned}
&\Delta W_j \mathbf{I}_{j,j} + \Delta W_{j+1}\mathbf{I}_{j,j+1} + \Delta W_{j+2}\mathbf{I}_{j,j+2}\\
&\quad = \sum_{i=j}^{j+2} N_{i,3}(t_{pj})W_i(\mathbf{P}_j - \mathbf{C}(t_{pj}))
\end{aligned}
$$

where

$$\mathbf{I}_{j,i} = N_{i,3}(t_{pj})(\mathbf{B}_i - \mathbf{P}_j)$$

Express the above equation by the following form:

$$
\mathbf{F}_1 + \mathbf{F}_2 + \mathbf{F}_3 = \sum_{i=j}^{j+2} N_{i,3}(t_{pj})W_i\Delta\mathbf{C}(t_{pj})
\tag{9}
$$

where

$$\mathbf{F}_{i-j+1} = N_{i,3}(t_{pj})\Delta W_i(\mathbf{B}_i - \mathbf{P}_j)$$

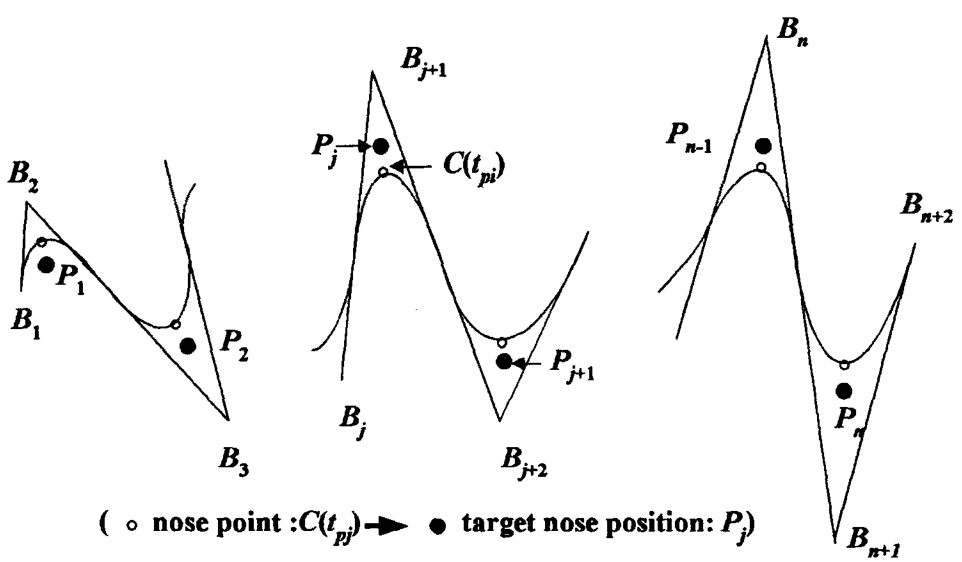

Figure 5 Curve-fitting improvement by moving nose points.

for $i = j, j+1, j+2$

By referring to equation (9), the curve point modification can be given the following physical meaning: the point P_j is pulled by three forces, yielding displacement $\Delta\mathbf{C}(t_{pj})$ as illustrated in *Figure 4*.

Let $\mathbf{u}_j$ be the unit tangent vector of $\mathbf{B}_j+1-\mathbf{P}_j$, v_j be the unit tangent normal to $\mathbf{u}_j$, and each nose point $\mathbf{C}(t_{pj})$ be selected on line $\overrightarrow{\mathbf{P}_j\mathbf{B}_{j+1}}$, a scalar expression in $\mathbf{u}_j$-direction for equation (9) can be written below:

$$\Delta W_j J_{j,j} + \Delta W_{j+1} J_{j,j+1} + \Delta W_{j+2} J_{j,j+2} = \sum_{i=j}^{j+2} N_{i,3}(t_{pj}) W_i \Delta C(t_{pj}) \quad (10)$$

where $J_{j,i} = \mathbf{u}_j \cdot \mathbf{I}_{j,i}$

$\Delta C(t_{pj}) = \mathbf{u}_j \cdot \Delta\mathbf{C}(t_{pj})$

Express equation (9) by a scalar equation in v_j-direction:

$$(N_{j,3}(t_{pj})\Delta W_j)|(\mathbf{B}_j-\mathbf{P}_j)\times\mathbf{u}_j| = (N_{j+2,3}(t_{pj})\Delta W_{j+2})|(\mathbf{B}_{j+2}-\mathbf{P}_j)\times\mathbf{u}_j| \quad (11)$$

Rewrite the above equation into the following form

$$\frac{\Delta W_{j+2}}{\Delta W_j} = \frac{|(\mathbf{B}_j-\mathbf{P}_j)\times\mathbf{u}_j|}{|(\mathbf{B}_{j+2}-\mathbf{P}_j)\times\mathbf{u}_j|}\frac{N_{j,3}(t_{pj})}{N_{j+2,3}(t_{pj})} \quad (12)$$

To get true solutions of weight changes ΔW_i the solutions obtained by equation (10) should be equal to that by equation (12). But in fact, the two sets of solutions are always different, and one cannot solely use equation (10) or equation (12) to reach a satisfactory result for curve modification. Sections 2.3.1, 2.3.2 and 2.3.3 will further discuss the solutions of weight changes based on equations (10) and (12).

Solutions of ΔW_i in $\mathbf{u}_j$-direction. In this paper, there are two constraints to modify a NURBS curve by moving the nose point features: (1) there is only a single curve point $\Delta\mathbf{C}(t_{pj})$ to be modified for each curve segment, and the curve point can be viewed as the nose point of this segment; (2) weights for the two end control points always remain constant, i.e. $\Delta W_1 = \Delta W_{n+2} = 0$. From the above two constraints, n equations with n unknown weight changes ΔW_i expressed by equation (10) can be written as the following matrix form:

$$\begin{bmatrix} J_{1,2} & J_{1,3} & 0 & & \cdots & & 0 \\ J_{2,2} & J_{2,3} & b_{2,4} & 0 & & & 0 \\ 0 & & & 0 & & & \\ & 0 & \ddots & & \ddots & & \vdots \\ \vdots & & & \ddots & J_{j,j-1} & J_{j,j} \quad J_{j,j+1} & \\ & & & & \ddots & & 0 \\ & & & & 0 & J_{n-1,n-1} \quad J_{n-1,n} & J_{n-1,n+1} \\ 0 & & \vdots & & & 0 \quad J_{n,n} & J_{n,n+1} \end{bmatrix} \begin{bmatrix} \Delta W_2 \\ \Delta W_3 \\ \\ \\ \\ \\ \Delta W_n \\ \Delta W_{n+1} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n+2} N_{i,3}(t_{p1}) W_i \Delta C(t_{p1}) \\ \sum_{i=1}^{n+2} N_{i,3}(t_{p2}) W_i \Delta C(t_{p2}) \\ \vdots \\ \sum_{i=1}^{n+2} N_{i,3}(t_{pj}) W_i \Delta C(t_{pj}) \\ \vdots \\ \sum_{i=1}^{n+2} N_{i,3}(t_{pn}) W_i \Delta C(t_{pn}) \end{bmatrix} \quad (13)$$

The new weights should always remain positive for modifying a NURBS curve.

However, weight changes ΔW_i calculated from equation (13) may be either positive or negative. *Figure 8*(a) shows the initial curve and the two target nose points of a face profile. Assume all the initial weights are equal to 1, the weight changes from equation (13) are given

$$\Delta W = [4.680\ 2.967\ -0.400\ -2.418\ -0.757\ 0.847\ -0.693\ -0.937\ -0.923\ -0.302\ 0.189\ -0.470\ -0.793\ -0.522\ -0.467\ -0.236] \quad (13)$$

The new weight for the fourth control point will be negative (-1.418), which is invalid for a NURBS curve. If any of the new weight given from equation

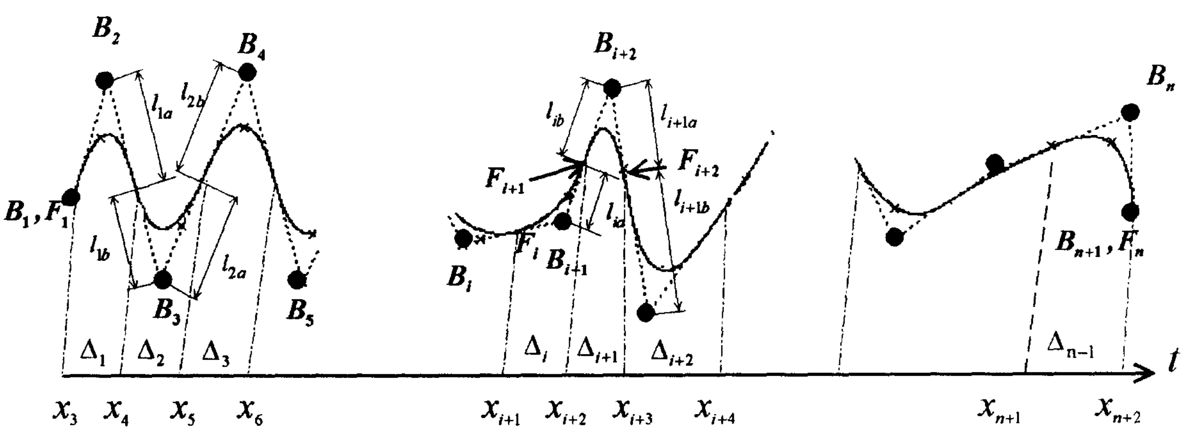

Figure 6 A force system of three forces acting on the target point $\mathbf{P}_j$

(13) is negative, additional allocation for weight changes is needed, which will be discussed below.

Assume the nose point has moved to the target position along the $\mathbf{u}_j$-direction by weight changes using equation (13). Considering the allocation of weight changes should not destroy what movement has been obtained by equation (13). Let $\Delta W_j'$ be the difference of ΔW_j, the following equation can be derived from equation (10) to express the relationship between the differences of weight changes.

$$\Delta W'_{i-1}J_{i-1,i-1}+\Delta W'_iJ_{i-1,i}+\Delta W'_{i+1}J_{i-1,i+1}=0 \qquad (14)$$

where $i=2, 3, \ldots, n+1$. Process 1 is an algorithm to allocate the weight changes all over the curve segments, until all of the new weights become positive. After the allocation of weight changes by the

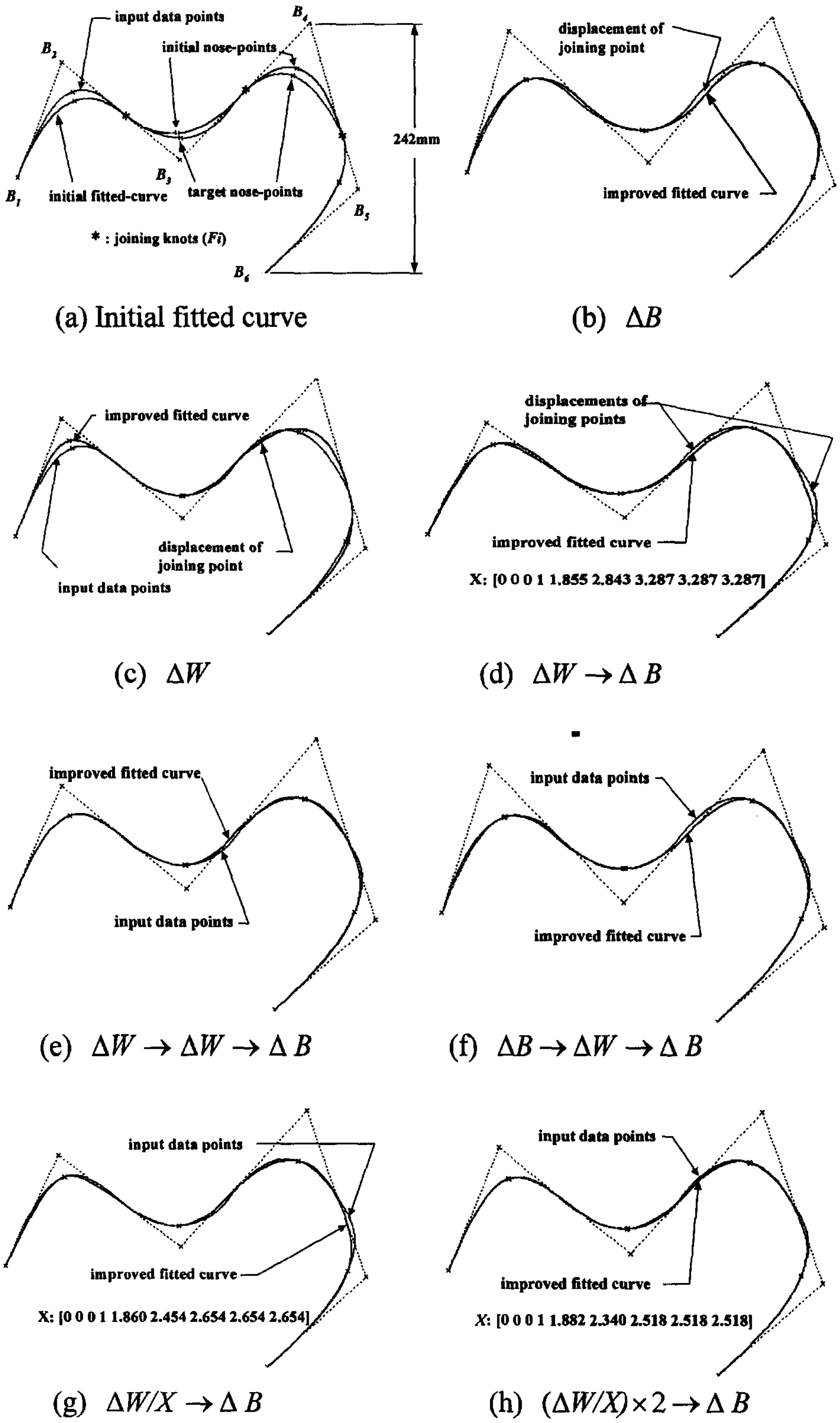

Figure 7 Curve modification in need of allocation of weight changes.

[Process 1]:

```
For(i = 2; i <= n+1; i++)
{
    If : ΔW_i < -W_i (i.e. W_i = ΔW_i + W_i < 0)     } if the weight for nose control point of segment # i-1
                                                       will become negative according to equation (13)
    Then : {
        ΔW'_i = -ΔW_i,
        For(;;)                                        allocate difference of ΔW for segment # i-1
        {                                              (ΔW'_{i-1}, ΔW'_i, ΔW'_{i+1}) according to
         Assign values A and B;                        equation (14), where A+B=1
         ΔW'_{i+1} = -A (J_{i-1,i} / J_{i-1,i+1}) ΔW'_i;
         ΔW'_{i-1} = -B (J_{i-1,i} / J_{i-1,i-1}) ΔW'_i;

        If : ((W_{i-1} + ΔW_{i-1}) > 0 and (W_{i+1} + ΔW'_{i+1}) > 0)   } make sure the weight changes for
        Then : Break;                                                     segment i-1 are positive
        }
        For(j = i-1; j-1 >= 1; j--)                                       } make ΔW' for segments #i-2, i-3, ..., 1
         ΔW'_{j-1} = -(J_{j-1,j} / J_{j-1,j-1}) ΔW'_j - (J_{j-1,j+1} / J_{j-1,j-1}) ΔW'_{j+1};   to be congruent with equation (14)
        For(j = i+1; j+1 <= n+1; j++)                                     } make ΔW' for segments #i, i+1, ..., n
         ΔW'_{j+1} = -(J_{j-1,j-1} / J_{j-1,j+1}) ΔW'_{j-1} - (J_{j-1,j} / J_{j-1,j+1}) ΔW'_j;   to be congruent with equation (14)
        For(j = 1; j <= n+2; j++)                                         } compute weight changes
         ΔW = ΔW_j + ΔW'_j;
    }
}
```

algorithm of Process 1, equation (10) will still keep true, but the curve point will deflect slightly in the v_j-direction. When the algorithm of Process 1 is applied to the problem in *Figure 8*(a) to allocate the weight changes, the weight changes become

$$\Delta W = [9.417\ 5.333\ -0.400\ 0.000\ -0.176\ 3.264\ -0.097\ -0.152\ 1.033\ -0.276\ -0.852\ 0.724\ 1.411\ -0.923\ -0.072\ 0.157]$$

The result for curve modification is shown in *Figure 8*(b).

Solutions of ΔW_i for both $\mathbf{u}_j$- and $\mathbf{v}_j$-directions. From equations (10) and (12), one can understand that the solutions derived from these two equations are incompatible with each other for multiple-point modifications. Since in this case, the nose points do not move purely in (or opposite to) the $\mathbf{u}_j$-direction. Instead, they may also move in the v_j-direction concurrently when changing the weights. Thus, one cannot move the nose points to the target positions, except if the user can assign the nose points and the target positions so that the following equation can be satisfied:

$$W_j \mathbf{I}_{j,j} \cdot v_j + W_{j+2} \mathbf{I}_{j,j+2} \cdot v_j = 0 \tag{15}$$

Figure 7(c) shows the result of curve modification by changing weights. It appears that using the weight changes obtained by equation (13) can only move the nose points closer to target positions, which is because of the effect in the v_j-direction. If the nose point movements are evident in the v_j-direction, two approaches for further adjustment to give better results are suggested: (1) move nose points by changing knot vectors so as to meet equation (15), and (2) repeat equation (13) followed by Process 1 until an optimal solution is reached. The lists in *Table 1* show that the precision of the results of nose point movements are raised as the change of weights by equation (13) are repeated, but the improvement of precision is not apparent after the fourth weight change.

Local modification for a single curve segment. If it is necessary to modify a single curve segment, e.g. curve segment i, by moving the nose point toward the target position, both equations (10) and (12) should be satisfied for weight changes in the meantime. First, set the weight change ΔW_{i+1} for the nose control point to a constant, and allocate the two weight changes ΔW_i and ΔW_{i+2} by combining both equations (10) and (12). These two equations are combined into the following form:

$$\Delta W_i J_{i,i} + \Delta W_{i+2} J_{i,i+2} = \Delta W_i^* J_{i,i} + \Delta W_{i+2}^* J_{i,i+2} = \Delta W_i^* (J_{i,i} + d_i J_{i,i+2})$$

where ΔW^* is the new value of ΔW, and

$$d_i = \frac{|V_{i,i} \times u_i|}{|V_{i,i+2} \times u_i|} \frac{N_{i,3}(t_{pi})}{N_{i+2,3}(t_{pi})} \tag{16}$$

Finally, the governing weight changes for the curve segment become

Table 1 **Results of curve modification for five times of ΔW**

	W: weight vector Δ*W*: weight change vector	*Err*: error vector of nose points
Initial model	*W*:[1 1 1 1 1 1]	[8.960, 5.550, 8.275, 0.396]
1 Weight change	Δ*W*:[0 5.170 5.198 2.717 1.778 0] *W*:1 6.170 6.198 3.717 2.778 1]	[7.563, 1.513, 6.535, 2.591]
2 Weight changes	Δ*W*:[0 −3.356 −3.501 −2.159 −1.329 0] *W*:[1 2.814 2.697 1.558 1.449 1]	[1.860, 1.360, 4.780, 1.138]
3 Weight changes	Δ*W*:[0 −0.394 −0.384 −0.323 −0.239 0] *W*:1 2.420 2.313 1.326 1.210 1]	[1.280, 1.327, 4.850, 0.204]
4 Weight changes	Δ*W*:[0 −0.006 −0.013 −0.012 −0.011 0] *W*:[1 2.414 2.300 1.314 1.199 1]	[1.280, 1.329, 4.851, 0.197]
5 Weight changes	Δ*W*:[0 0.001 0.002 0.001 0 0] *W*:[1 2.415 2.302 1.315 1.199 1]	[1.287, 1.315, 4.862, 0.184]

$$\Delta W_i^* = \frac{\Delta W_i J_{i,i} + \Delta W_{i+2} J_{i,i+2}}{J_{i,i} + d_i J_{i,i+2}} \quad (17)$$

$$\Delta W_{i+1}^* = \Delta W_{i+1} \quad (18)$$

$$\Delta W_{i+2}^* = d_i \Delta W_i^* \quad (19)$$

The local curve modification of solely moving the nose point of a curve segment can be achieved by firstly solving the weight changes using equation (10), and next adjusting the weight changes for the local segment using equations (17)–(19).

Change both weights and control points

To improve curve fitting, one may manage to improve nose points by moving control points and changing weights in combination. The cases from *Figure* 7(d–f) depict the results by combining control point movements and weight changes. In these cases, two weight changes followed by control point movement [as indicated by the symbol $\Delta W \rightarrow \Delta W \rightarrow \Delta B$ in *Figure* 7(e)], is found to render better precision for improving curve fitting.

Positioning of joining knots

When weights and control points are changed for curve modification, the curve segments will more or less deflect due to the displacement of joining points between every two adjacent curve segments, although the nose points get close to the target positions. The deflection of curve segments may reduces the performance of curve modification, and should be decreased. *Figure* 7(b) and (d) reveal the displacement of joining points resulting from moving control points and changing weights, which destroys the improvement of curve fitting near the joining points. *Figure 8*(b) shows a large amount of deflection in the curve segment beneath the chin, which is caused by the displacement of the joining point. The adoption of a new knot vector with reference to the following equation will moderate the deflections of curve segments by constraining the joining points to stay at their original places.

$$\Delta_{i+1} = \frac{W_{i+2}(B_{i+2} - F_{i+1})}{W_{i+1}(F_{i+1} - B_{i+1})} \Delta_i \quad (20)$$

where Δ_j is the knot span, B_j represents the control point, and F_j is the joining point between two curve segments (see *Figure 6*), and equation (20) has been proved in the Appendix. Set the first span Δ_1 to a constant and calculate other spans accordingly:

$$\Delta_i = \frac{W_{i+1}}{W_2}\left(\frac{B_{i+1} - F_i}{F_i - B_i}\right)\cdot\left(\frac{B_i - F_{i-1}}{F_{i-1} - B_{i-1}}\right)\cdot\ldots$$

$$\times\left(\frac{B_4 - F_3}{F_3 - B_3}\right)\cdot\left(\frac{B_3 - F_2}{F_2 - B_2}\right)\cdot\Delta_1 \quad (21)$$

for $i \geq 2$. The knot vector can be expressed as:

$$[X] = [0\ 0\ 0\ \Delta_{1,1}\ \Delta_{1,2}\ \Delta_{1,3}\ \ldots\ \Delta_{1,i}\ \ldots\ \Delta_{1,n-2}\ \Delta_{1,n-1}\ \Delta_{1,n-1}\ \Delta_{n,n-1}]$$

where $\Delta_{i,j} = \Delta_1 + \Delta_2 + \ldots + \Delta_{i-1} + \Delta_i$.

When constraining the joining points of curve segments to the weight changes, *Figure* 7(d) becomes *Figure* 7(g) and *Figure* 8(e) becomes *Figure* 8(h), which significantly decrease the deflection of curve segments near the joining points. Whereas, when constraining the joining points by changing the knot vector, the nose point positions will move in the meantime, and the operations of moving control points and/or changing weights must be applied again to improve the curve shape. Therefore, to modify a NURBS curve by moving nose points, a series of operations for locating nose points and joining points should proceed until an optimal precision can be reached.

Assume $\Delta W/X$ represents the constraining new knot vector to the change of weights, so as to locate the joining points, and that moderates the curve deflections of curve segments. *Table 2* lists the results of curve-fitting improvements during five times of $\Delta W/X$, and demonstrates that the precision of curve modification rises as the number $\Delta W/X$ increases. From *Figure* 7(g) and (h) and *Table 2*, it is revealed that constraining joining points to weight changes ($\Delta W/X$) apparently helps to improve the total precision of a fitted curve for the subsequent control point movements.

Summing up the above discussions for curve modification, five characteristics can be derived, as follows:

1. Moving control points gives fast and precise location of nose points, while the relevant joining knots of curve segments will move, and the curve segments may deflect.
2. Weight changes for multiple nose point modifications will make the result of movement less precise, while, repetitive weight changes, several times, will cause little joining knot displacements, and will give better global curve modification for nose points.
3. The precision of curve modification by weight changes may be significantly increased when constraining the joining knots.
4. The precision of curve modification significantly depends upon the vectors from the initial nose points toward target positions. When equation (15) is satisfied, an optimal result is yielded.
5. A number of changes of weights and knot vector followed by control point movements may achieve better and quicker curve modifications.

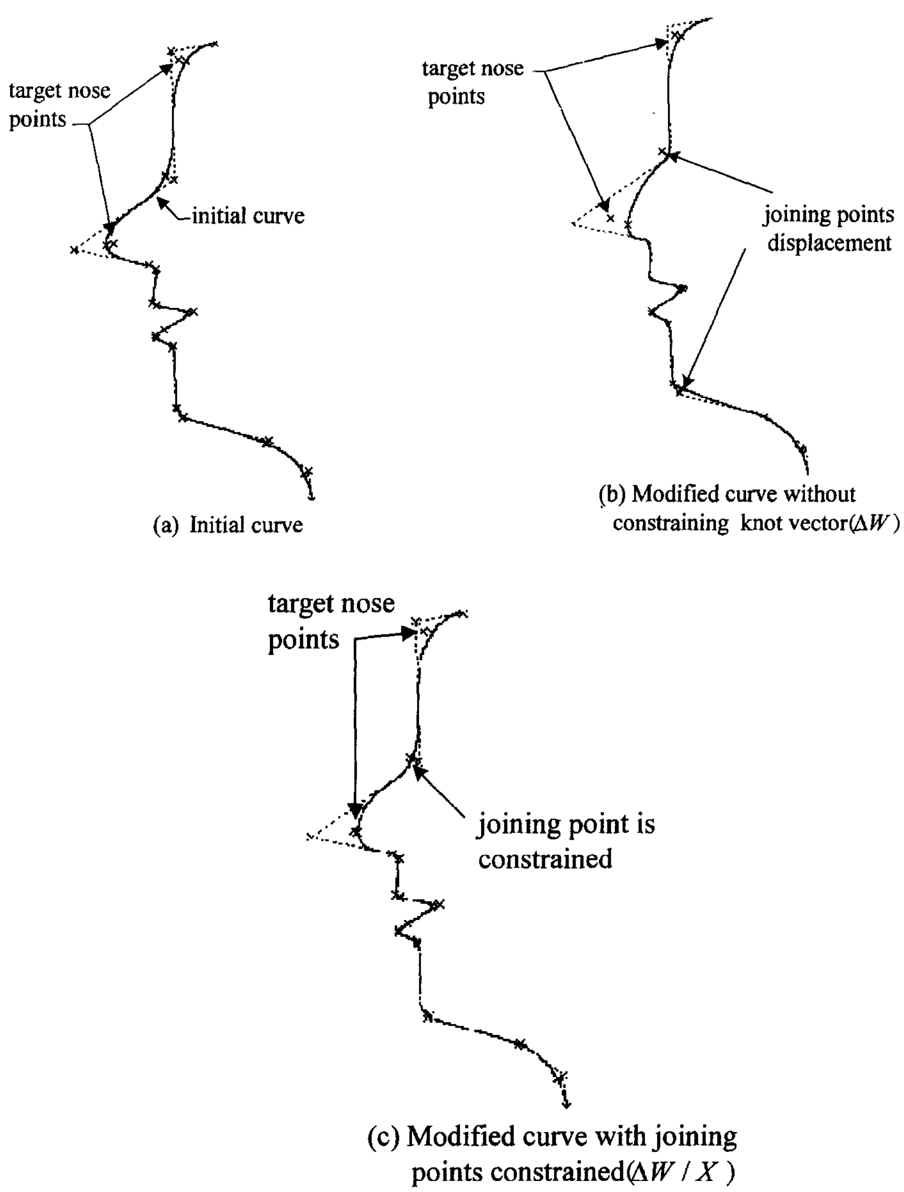

Figure 8 The joining points F_i and knot spans Δ_i relationship.

Table 2 Results of curve modification for five times of $\Delta W/X$

	***W*: weight vector** **Δ*W*: weight change vector**	***X*: knot vector** ***Err*: error vector of nose points**
Initial fitted curve	*W*:[1 1 1 1 1 1]	*X*:[0 0 0 1 1.855 2.843 3.287 3.287 3.287] *Err*:[8.960, 5.550, 8.275, 0.396]
1 Weight change	Δ*W*:[0 5.170 5.198 2.717 1.778 0] *W*:1 6.170 6.198 3.717 2.778 1]	*X*:[0 0 0 1 1.860 2.454 2.654 2.654 2.654] *Err*:[7.584, 4.038, 7.932, 4.507]
2 Weight changes	Δ*W*:[0 −3.218 −3.154 −2.348 −1.595 0] *W*:[1 2.952 3.044 1.369 1.183 1]	*X*:[0 0 0 1 1.882 2.340 2.518 2.518 2.518] *Err*:[1.779, 3.119, 5.595, 0.734]
3 Weight changes	Δ*W*:[0 −0.386 −0.437 −0.407 −0.271 0] *W*:[1 2.566 2.607 0.962 0.912 1]	*X*:[0 0 0 1 1 868 2.239 2.397 2.397 2.397] *Err*:[1.384, 2.069, 4.262, 0.690]
4 Weight changes	Δ*W*:[0 −0.067 −0.139 −0.156 0.073 0] *W*:[1 2.499 2.468 0.806 0.839 1]	*X*:[0 0 0 1 1.845 2.163 2.312 2.312 2.312] *Err*:[1.391, 2.473, 3.305, 0.903]
5 Weight changes	Δ*W*:[0 −0.006 −0.039 −0.025 0.011 0] *W*:[1 2.470 2.377 0.724 0.820 1]	*X*:[0 0 0 1 1.823 2.113 2.260 2.260 2.260] *Err*:[1.402, 2.402, 2.682, 0.903]

From the above observations, a number of joining points constrained to weight changes $\Delta W/X$ followed by control point movements ΔB can achieve higher precision for curve modification. The global precision for the curve modifications or curve-fitting improvements will rise as the number n of the change $\Delta W/X$ increases initially, but the efficiency of the global precision improvements will slow down (even be destroyed) as n keeps increasing. As *Table 2* shows, the precision improvement of nose points almost ceases from the fourth time of $\Delta W/X$. An optimal solution can be reached by choosing the value n when the slow down of the precision improvement reaches a certain scale. Set the nose point errors *Err*: $[e_{i,1}, e_{i,2}, \ldots, e_{i,n}]$, the initial nose point errors *Err*: $[e_{0,1}, e_{0,2}, \ldots, e_{0,n}]$, and the error improvements ΔErr: $[(e_{i,1}, \Delta e_{i,2}, \ldots, \Delta e_{i,n}]$; the improvement efficiency is defined as:

$$\rho = \frac{\Delta e_{i,1} + \Delta e_{i,2} + \ldots + \Delta e_{i,n}}{e_{i-1,1} + e_{i-1,2} + \ldots + e_{i-1,n}} \tag{22}$$

where $\Delta e_{i,j} = e_{i-1,j} - e_{i,j}$

When the improvement efficiency ρ has been less than a certain value or even become negative, stop

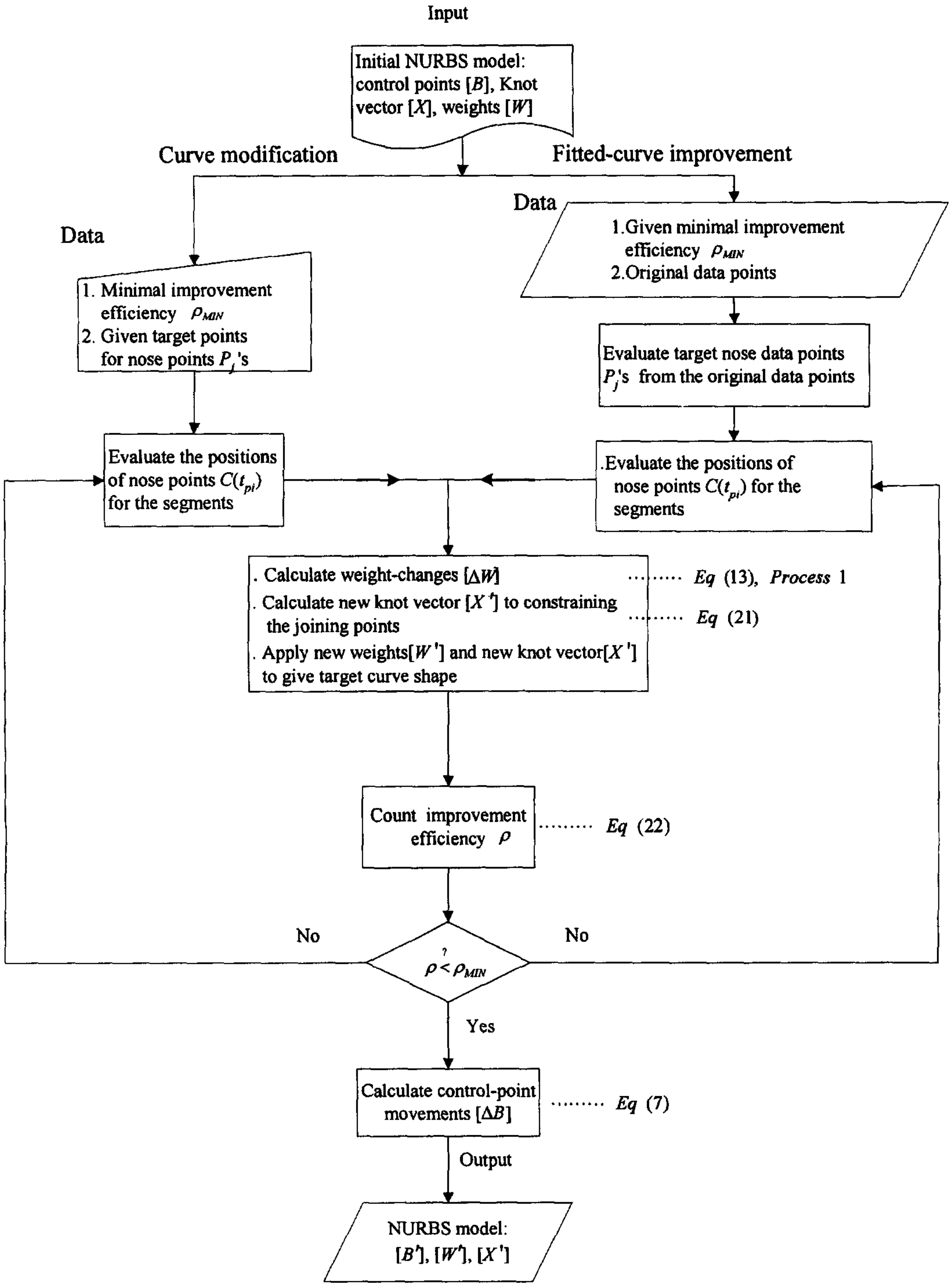

Figure 9 Flow chart for NURBS curve modification.

the $\Delta W/X$ process and proceed with the control point movements, to give an optimal modification for the NURBS curve.

The recipe for curve modifications or curve-fitting improvements, covering the algorithms proposed in this paper, is illustrated in *Figure 9*. The following implementation first gives examples which get closer to implementing the behaviors of weight changes and control point movements, and subsequently demonstrates the results of the proposed optimal criteria.

Computer implementation

In order to put the curve-modification strategy into practice, this research employs C++ programming language to develop a prototype system, which

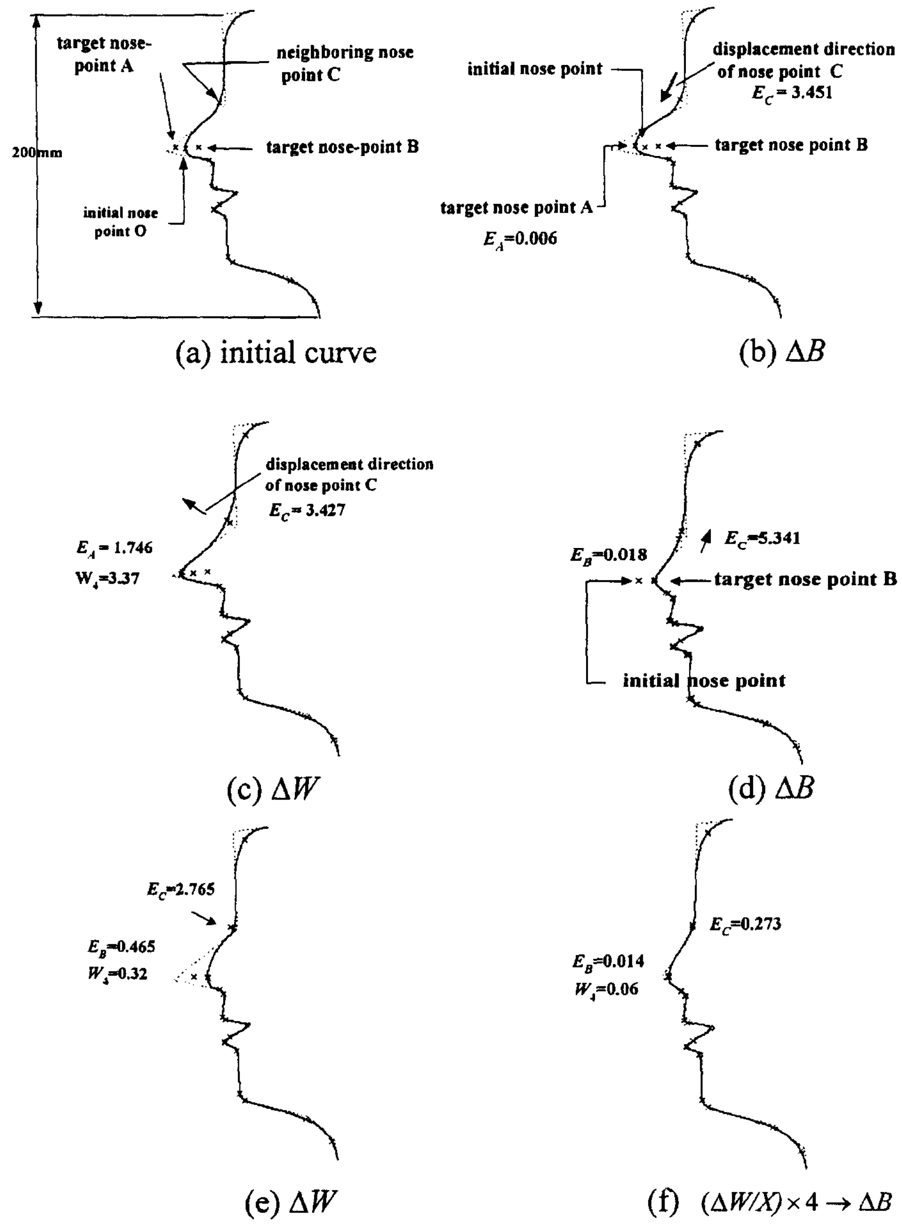

Figure 10 Curve modifications of a face profile.

Table 3 Improvement efficiencies of $(\Delta W/X)\times 4$ (mm)

No. of $\Delta W/X$ modification	E_B	E_C	ρ
Modification for 1st time of $\Delta W/X$	0.456	2.765	0.973
Modification for 2nd time of $\Delta W/X$	0.218	0.542	0.731
Modification for 3rd time of $\Delta W/X$	0.183	0.364	0.280
Modification for 4th time of $\Delta W/X$	0.147	0.269	0.218

accounts for performing the algorithms proposed in this paper. The prototype system was developed using a Pentium-166 personal computer with 64 Mbytes of RAM. The computer platform is Windows 95 and the programming language is Borland C++.

Figure 10(a) shows the initial curve of a face profile. The original nose tip **O** is to be moved toward target points **A** and **B**. A neighboring nose point **C** is used to help evaluate the improvement efficiency.

Figure 10(b) and (d) show the nose point modifications toward points **A** and **B** respectively which resulted from moving the control point using equation (7). *Figure 10*(c) and (e) depict the nose point modifications toward points **A** and **B** respectively which resulted from changing the weights using equation (13), with all the initial weights $W_i = 1$. Compare *Figure 10*(b) and (c) [or compare *Figure 10*(d) and (e)], which reveals that weight changes result in a larger distance error between the neighboring nose point **C** to the resulting curve than the movement of control points does. This is because the direction of displacements for the two cases are different and almost normal to each other.

The distance error caused by changing weights can be eliminated as the joining points are constrained. Assume the allowable displacement for the neighboring nose point **C** is 0.3 mm, and the minimal improvement efficiency ρ_{MIN} is set to 0.25. The improvement efficiency values ρs for moving nose point **B** toward target position in the first, second, third and fourth times $\Delta W/X$ are illustrated in *Table 3*. ρ is smaller than ρ_{MIN}, and the displacement of neighboring nose point C is smaller than the allowable value in the fourth time of $\Delta W/X$. Thus four times of the changes of weights and knot vector followed by control point movements, i.e. $\Delta W/X) \times 4 \rightarrow \Delta \mathbf{B}$ as indicated in the figure, gives an optimal result, which yields nose point error 0.014 mm and displacement of the neighboring nose point 0.273 mm [see *Figure 10* (f)].

Conclusions

In this paper, the nose points and joining points on curve segments are viewed as feature points. Local modification of a NURBS curve is to be accomplished by fitting the curve to these feature points. Some algorithms are proposed to determine the change of weights, the movement of control points, and the constraint of knot vector. The proposed algorithms are used either to improve the accuracy of a fitted curve or to modify an existing curve. An equilibrium equation of three forces acting on each nose point to form the displacement of the nose point is created to physically explain the solution of weight changes, and with which, a weight change allocation process is proposed to transform weights with negative values (if any) into positive ones so that NURBS curve modification can be undertaken. Constraining both nose points and joining points for curve segments keep curve segments from excessive deflection, and can give better results for curve modification. A prototype computer system is also developed to verify the usefulness of the proposed methodology. The implementation examples discussed in the paper lead to the conclusion that several times of weight changes and knot point constraining followed by control point movements can give an optimal result.

References

1 **Taura, T, Nagasaka, I and Yamagishi, A** 'Application of evolutionary programming to shape design', *Computer Aided Design* Vol 30 No 1 (1998) 29–35
2 **Piegl, L** 'On NURBS: a survey', *IEEE Computer Graphics and Applications* Vol 11 No 1 (1991) 55–71
3 **Piegl, L** 'Modifying the shape of rational B-splines. Part 1: curves', *Computer Aided Design* Vol 21 No 8 (1989) 509–518
4 **Fowler, B and Bartels, R** 'Constraint-based curve manipulation', *IEEE Computer Graphics and Applications* Vol 13 No 5 (1993) 43–49
5 **Au, C K and Yuen, M M** 'Unified approach to NURBS curve shape modification', *Computer Aided Design* Vol 27 No 2 (1995) 85–93
6 **Sànchez-Reyes, J** 'A simple technique for NURBS shape modification', *IEEE Computer Graphics and Applications* Vol 17 No 1 (1997) 52–59
7 **Rogers, D F and Adams, J A** *Mathematical Elements for Computer Graphics*, 2nd edn. McGraw-Hill, New York, USA, 1990.
8 **Prautzsch, H** 'Degree elevation of B-spline curves', *Computer Aided Geometric Design* Vol 1 (1984) 193–198
9 **Boehm, W** 'On the efficiency of knot insertion algorithms', *Computer Aided Geometric Design* Vol 2 (1985) 141–143
10 **Boehm, W** 'Inserting new knots into B-spline curves', *Computer Aided Design* Vol 12 No 4 (1980) 199–201
11 **Choi, B K, Yoo, W S and Lee, C S** 'Matrix representation for NURB curves and surfaces', *Computer Aided Design* Vol 22 No 4 (1990) 235–240

Appendix

For a third order NURBS curve[7,11], a curve segment can be expressed as:

$$C^i(t) = \sum_{m=i}^{i+2} \frac{N_{m,3}(t)W_m}{\sum_{m=i}^{i+2} N_{m,3}(t)W_m} B_m \qquad \text{(A1)}$$

the basis functions are formulated as:

$$N_{m,k}(t) = \frac{(t-x_m)N_{m,k-1}(t)}{x_{m+k-1}-x_m} + \frac{(x_{m+k}-t)N_{m+1,k-1}(t)}{x_{m+k}-x_{m+1}}$$

$$= \frac{(t-x_m)N_{m,k-1}(t)}{\Delta_{m+k-4}+\Delta_{m+k-5}+\ldots+\Delta_{m-2}}$$

$$+ \frac{(x_{m+k}-t)N_{m+1,k-1}(t)}{\Delta_{m+k-3}+\Delta_{m+k-4}+\ldots+\Delta_{m-1}} \qquad \text{(A2)}$$

$$N_{m,1}(t) = \begin{cases} 1 \text{ for } m = i+2 \\ 0 \text{ for other } m \end{cases} \qquad \text{(A3)}$$

For an n segment NURBS curve, the knot vector for open third order B-spline[7] has such a pattern:

$$[X] = [x_1\ x_2\ x_3\ \ldots\ x_{n+3}\ x_{n+4}\ n_{n+5}]$$
$$= [0\ 0\ 0\ x_4\ x_5\ \ldots\ x_{n+2}\ x_{n+3}\ x_{n+3}\ x_{n+3}] \qquad \text{(A4)}$$

From equations (A2) and (A3), the basis functions on knot x_{i+3} for parameter range $x_{i+2} \le t < x_{i+3}$:

$$N_{i,3}(x_{i+3}) = 0$$

$$N_{i+1,3}(x_{i+3}) = \frac{(x_{i+3} - x_{i+1})N_{i+1,2}(x_{i+3})}{\Delta_i + \Delta_{i-1}} + \frac{(x_{i+4} - x_{i+3})N_{i+2,2}(x_{i+3})}{\Delta_{i+1} + \Delta_i} \quad (A5)$$

$$N_{i+2,3}(x_{i+3}) = \frac{(x_{i+3} - x_{i+2})N_{i+2,2}}{\Delta_{i+1} + \Delta_i} \quad (A6)$$

$$N_{i+1,2}(x_{i+3}) = \frac{(x_{i+3} - x_{i+3})N_{i+2,1}}{\Delta_i} = 0 \quad (A7)$$

$$N_{i+2,2}(x_{i+3}) = \frac{(x_{i+3} - x_{i+2})N_{i+2,1}}{\Delta_i} = 1 \quad (A8)$$

where span $\Delta_j = x_{j+3} - x_{j+2}$. Substituting equations (A7) and (A8) for equations (A5) and (A6)

$$N_{i+1,3}(x_{i+3}) = \frac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i} \quad (A9)$$

$$N_{i+2,3}(x_{i+3}) = \frac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i} \quad (A10)$$

Express the joining point by substituting the above equations for equation (A1)

$$F_{i+1} = C^i(x_{i+3}) = \frac{N_{i+1,3}(x_{i+3})\,W_{i+1}}{\sum_{m=i}^{i+2} N_{m,3}(x_{i+3})W_m} B_{i+1} + \frac{N_{i+2,3}(x_{i+3})W_{i+2}}{\sum_{m=i}^{i+2} N_{m,3}(x_{i+3})W_m} B_{i+2}$$

$$= \frac{\dfrac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i}}{\sum_{m=i}^{i+2} N_{m,3}(x_{i+3})W_m} W_{i+1}B_{i+1} + \frac{\dfrac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i}}{\sum_{m=i}^{i+2} N_{m,3}(x_{i+3})W_m} W_{i+2}B_{i+2}$$

$$= \frac{\Delta_{i+1}}{\Delta_{i+1}W_{i+1} + \Delta_i W_{i+2}} W_{i+1}B_{i+1} + \frac{\Delta_i}{\Delta_{i+1}W_{i+1} + \Delta_i W_{i+2}} W_{i+2}B_{i+2} \quad (A11)$$

Rearrange equation (A11) into

$$\Delta_{i+1} = \frac{W_{i+2}(B_{i+2} - F_{i+1})}{W_{i+1}(F_{i+1} - B_{i+1})} \Delta_i \quad (A12)$$

Equation (A12) formulates the relationship among knot spans of a NURBS curve.

For a non-rational NUBS curve ($W_i = 1$), rather, the knot point (feature point) is formulated as:

$$F_{i+1} = C^i(x_{i+3}) = \frac{\Delta_{i+1}}{\Delta_{i+1} + \Delta_i} B_{i+1} + \frac{\Delta_i}{\Delta_{i+1} + \Delta_i} B_{i+2} \quad (A13)$$

Once Δ_i is given, Δ_{i+1} will be substantiated following

$$\Delta_{i+1} = \frac{B_{i+2} - F_{i+1}}{F_{i+1} - B_{i+1}} \Delta_i \quad (A14)$$

By equation (A13), we can determine the knot vector from the obtained polygon vertices and feature points. If we appoint the first span Δ_1 constant (say, $\Delta_1 = 1$), the other spans will be derived from equation (A12):

$$\Delta_2 = \frac{W_3(B_3 - F_2)}{W_2(F_2 - B_2)} \Delta_1, \Delta_3 = \frac{W_4(B_4 - F_3)}{W_3(F_3 - B_3)} \Delta_2 \ldots,$$

$$\Delta_i = \frac{W_{i+1}(B_{i+1} - F_i)}{W_i(F_i - B_i)} \Delta_{i-1}, \ldots \quad (A15)$$

A general solution for Δ_i can be written as

$$\Delta_i = \frac{W_{i+1}}{W_2} \left(\frac{B_{i+1} - F_i}{F_i - B_i} \right) \cdot \left(\frac{B_i - F_{i-1}}{F_{i-1} - B_{i-1}} \right) \cdots \cdot \left(\frac{B_4 - F_3}{F_3 - B_3} \right) \cdot \left(\frac{B_3 - F_2}{F_2 - B_2} \right) \cdot \Delta_1 \quad (A16)$$

for $i \ge 2$. The knot vector is designated:

$$[X] = [0\ 0\ 0\ \Delta_{1,1}\ \Delta_{1,2}\ \Delta_{1,3} \ldots \Delta_{1,i} \ldots \Delta_{1,n-2}\ \Delta_{1,n-1}\ \Delta_{1,n-1}\ \Delta_{i,n-1}] \quad (A17)$$

where $\Delta_{i,j} = \Delta_i + \Delta_2 + \ldots + \Delta_{j-1} + \Delta_j$.